Martin Hitz, Rudolf Melcher (Hrsg.)

AIM'07
Alternative Interaktionsmechanismen

Lakeside Science and Technology Park

Seminarkonferenz
15. Juni 2007

Seminar aus Interaktive Systeme
Alpen-Adria-Universität Klagenfurt
Sommersemester 2007

Vorwort

Konferenzseminare haben an der Informatik der Alpen-Adria-Universität Klagenfurt Tradition. Die Forschungsgruppe Interaktive Systeme hat sich dieser Tradition angeschlossen und über einige Jahre hinweg ihr eigenes Format entwickelt, das sich mittlerweile gut bewährt hat, aber naturgemäß weiterhin als *Work in Progress* verstanden wird. An dieser Stelle des Entwicklungsprozesses soll das aktuelle Lehrveranstaltungsformat festgehalten werden, um als wohldefinierter Ausgangspunkt für weitere Optimierungen zur Verfügung zu stehen.

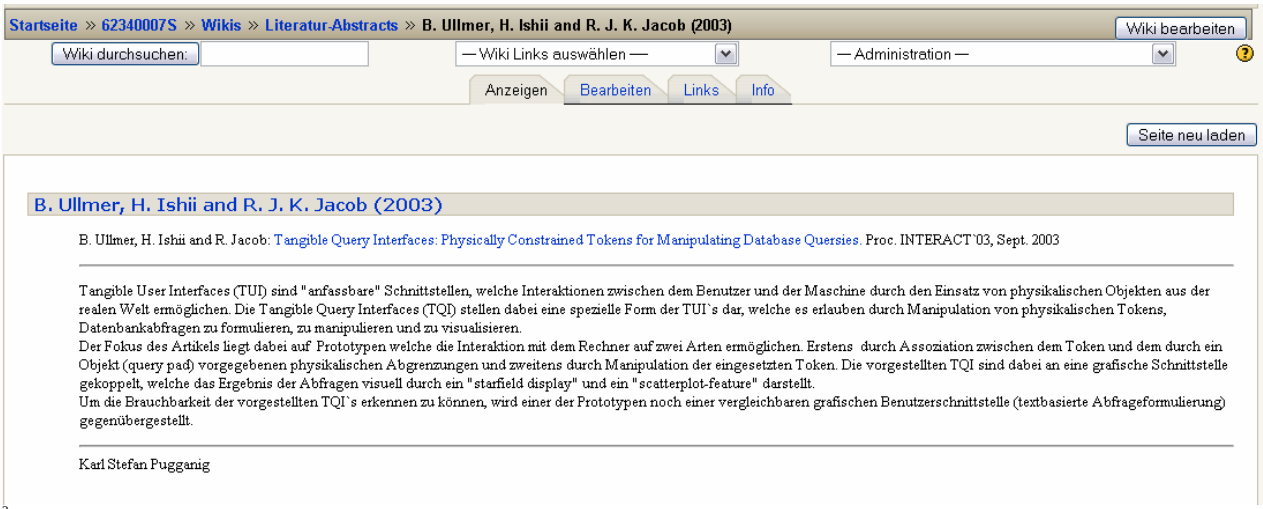
Ziel von Konferenzseminaren ist es, den Studierenden neben dem primären Training der Recherche und Rezeption von (i. A. englischsprachiger) wissenschaftlicher Originalliteratur auch den Prozess der Entwicklung eines Beitrags zu einer wissenschaftlichen Konferenz näher zu bringen. Zu diesem Zweck sind im Laufe des Semesters mehrere einschlägige Rollen zu probieren, wobei der Schwerpunkt auf der Rolle eines Autors bzw. einer Autorin einer eigenen (Überblicks-) Arbeit liegt.

Wir beginnen so früh wie möglich im Semester (SS 2007: 1. März) mit einer **Vorbesprechungseinheit**, in der das Rahmenthema sowie der Ablauf erläutert werden. Es wird ein kurzes Impulsreferat gehalten und ein *Call for Papers* besprochen, das – wie alle Lehrmaterialien – auf der Lehrplattform zu Verfügung gestellt wird. Das Impulsreferat wird durch eine Reihe von Einstiegsarbeiten (Originalartikel, evt. auch Lehrbuchkapitel) ergänzt, die ebenfalls auf der Lehrplattform verfügbar gemacht werden. Diese Basisliteratur soll den Studierenden einerseits einen Anker in die einschlägige Literatur, andererseits aber auch ein Muster für das Qualitätsniveau der als relevant erachteten Literatur bieten. Abgeschlossenen wird die Vorbesprechungseinheit mit der Vorstellung von Quellen und Werkzeugen zur Literaturrecherche sowie mit Hinweisen auf Zitierregeln und auf die allgemeine gute wissenschaftliche Praxis.

An diese Vorbesprechung schließt eine Phase der **Literaturrecherche** an (SS 2007: 1.-29. März), in der sich die Studierenden in das Gebiet einlesen und schließlich eine thematische Nische für ihre eigene Überblicksarbeit finden sollen. Dabei gilt, dass jede als relevant erachtete gelesene Arbeit auf der Lehrplattform zu hinterlegen ist, und zwar jedenfalls mit vollständiger Quellenangabe, mit einer *eigenen* Kurzfassung (1-2 Absätze) mit Hinweis auf für das Konferenzthema relevante Aspekte und, falls möglich, mit einem Link auf den Volltext der besprochenen Arbeit. Die so entstehende Literatursammlung steht allen Teilnehmern zur Verfügung und soll den Prozess der Literaturrecherche insgesamt beschleunigen[1]. Die wöchentlichen Plenareinheiten der Lehrveranstaltung dienen in dieser Phase zur Vorstellung solcher »Literaturfunde«: Die Teilnehmer erläutern in jeweils etwa 10 Minuten die Essenz des Inhalts eines von ihnen gelesenen und als besonders relevant eingestuften Artikel, wobei mindestens ein solcher Beitrag pro Person verpflichtend ist[2]. Aus diesen Diskussionen entstehen erste Ideen über die konkreten Beitragsthemen der Studierenden.

Diese Beitrags-Ideen werden in der folgenden **Konzeptionsphase** (SS 2007: 30. März – 19. April) konkretisiert, während der ein vorläufiger Titel und ein Abstract ausgearbeitet werden müssen, sowie mindestens drei

[1] Im SS 2007 wurde dafür ein Wiki definiert, das mit einem Beispieleintrag initialisiert wurde. Ein Eintrag eines Seminarteilnehmers ist nachstehend dargestellt. Über den Link in der Quellenangabe ist der Artikel im Volltext zu beziehen. Weiters war vorgesehen, über dasselbe Wiki auch ein Glossar wichtiger Begriffe zu etablieren (im Beispiel nicht ersichtlich). Es wurden zwischen 1 und 15 derartige Beiträge pro Person erfasst (Mittelwert knapp unter 5 / Person).


Startseite » 62340007S » Wikis » Literatur-Abstracts » B. Ullmer, H. Ishii and R. J. K. Jacob (2003)
Wiki bearbeiten
Wiki durchsuchen:
— Wiki Links auswählen —
— Administration —
Anzeigen
Bearbeiten
Links
Info
Seite neu laden
B. Ullmer, H. Ishii and R. J. K. Jacob (2003)
B. Ullmer, H. Ishii and R. Jacob: Tangible Query Interfaces: Physically Constrained Tokens for Manipulating Database Quersies. Proc. INTERACT'03, Sept. 2003
Tangible User Interfaces (TUI) sind "anfassbare" Schnittstellen, welche Interaktionen zwischen dem Benutzer und der Maschine durch den Einsatz von physikalischen Objekten aus der realen Welt ermöglichen. Die Tangible Query Interfaces (TQI) stellen dabei eine spezielle Form der TUI`s dar, welche es erlauben durch Manipulation von physikalischen Tokens, Datenbankabfragen zu formulieren, zu manipulieren und zu visualisieren.
Der Fokus des Artikels liegt dabei auf Prototypen welche die Interaktion mit dem Rechner auf zwei Arten ermöglichen. Erstens durch Assoziation zwischen dem Token und dem durch ein Objekt (query pad) vorgegebenen physikalischen Abgrenzungen und zweitens durch Manipulation der eingesetzten Token. Die vorgestellten TQI sind dabei an eine grafische Schnittstelle gekoppelt, welche das Ergebnis der Abfragen visuell durch ein "starfield display" und ein "scatterplot-feature" darstellt.
Um die Brauchbarkeit der vorgestellten TQI`s erkennen zu können, wird einer der Prototypen noch einer vergleichbaren grafischen Benutzerschnittstelle (textbasierte Abfrageformulierung) gegenübergestellt.
Karl Stefan Pugganig

[2] Auf Grund der i. A. stattfindenden Diskussion sind je 90 Minuten etwa sechs solcher Beiträge unterzubringen.

»Forschungsfragen«, die im Rahmen der zu erstellenden Arbeit beantwortet werden sollen. Gleichzeitig wird die Literaturarbeit (zielgerichtet) fortgesetzt. Am Ende dieser Phase werden die Abstracts an alle verteilt und die Themenstellungen in einer Plenarsitzung abgeglichen und endgültig festgelegt.

Nun folgt die **Ausarbeitungsphase** (SS 2007: 20. April – 9. Mai), in der die Erstfassungen der Beiträge erstellt werden. Die Literaturarbeit wird weiter fortgesetzt. Am Ende dieser Phase liegen alle Erstfassungen als (formatkonforme) PDF-Dateien auf der Lehrplattform vor.

In einer Plenarsitzung zu Beginn der **Begutachtungsphase** (SS 2007: 10.-24. Mai) werden für jede Arbeit drei Gutachter bzw. Gutachterinnen festgelegt und die Kriterien für ein konstruktives Gutachten vorgestellt, unterstützt durch ein reales Beispiel (5 Gutachten aus einem Begutachtungsprozess eines Konferenzbeitrags eines Mitglieds der Forschungsgruppe) und eine Erfassungsschablone für die numerische Beurteilung einer Reihe von Standardkriterien. Abgesehen von der Bewertung von diesen Standardkriterien sind die studentischen Gutachterinnen und Gutachter angehalten, eine Gesamtempfehlung abzugeben, und zwar durch Klassifikation der ihnen zugeordneten Beiträge in *Work in Progress* Arbeiten (»noch nicht ganz ausgereift«) und *Full Papers* (»ordentliche Publikation«). Da diese Klassifikation letztlich eine (wenn auch sehr schwache) Auswirkung auf die notenmäßige Beurteilung des Autors bzw. der Autorin hat[3], ist diese Phase gruppendynamisch relativ anspruchsvoll. Die endgültige Klassifikation der Arbeiten erfolgt in einer abschließenden Plenarsitzung (*Program Committee Meeting*), in der die einzelnen (auf der Lehrplattform abgelegten) Gutachten von den Gutachterinnen und Gutachtern vorgestellt und diskutiert werden.

Die **Finalisierungsphase** (SS 2007: 24. Mai – 13. Juni) dient zur Überarbeitung des eigenen Beitrags und zur Erstellung der *Camera Ready Copy*. Neben der verbesserten Arbeit ist eine kurze Stellungnahme abzugeben, in welcher Weise auf die Vorschläge der Gutachten eingegangen wurde. Die auf die Lehrplattform hochgeladene Endversion der Arbeit (PDF) wird von der Seminarleitung (mit Deckblatt, Seitennummern, Inhaltsverzeichnis, Kopf- und Fußzeilen ausgestattet) in einen Konferenzband übernommen, von dem zum Konferenztermin vorab für jeden Teilnehmer ein Exemplar vorbereitet wird.

Die **Präsentationsphase** (SS 2007: 15. Juni 8:30-17:30) entspricht dem simulierten Kongress. Dieser findet typischerweise außerhalb der üblichen Lehrveranstaltungsräumlichkeiten statt (SS 2007: *Lakeside* Demoraum, B01) und wird mit Pausenverpflegung und –getränken sowie einem gemeinsamen Konferenzessen »garniert«. Die Präsentationen sind mit 20 Minuten (*Full Paper*) bzw. 15 Minuten (*Work in Progress*) limitiert, dazu kommen Zeit für Diskussion und Vortragendenwechsel, sodass pro Beitrag 30 Minuten Bruttozeit zu veranschlagen sind. Die Teilnehmerinnen und Teilnehmer werden angehalten, über die drei besten Präsentationen abzustimmen – die Preisverleihung des *Best Presentation Awards* beschließt das Seminar.

Vorbereitete Materialien

Die Lehrplattform wird von der Seminarleitung mit folgenden Artefakten initialisiert:

- Strikte Formatvorlagen für LaTex und Word (nach den ACM SIGCHI Publikationsvorlagen)
- Quellen zur Literaturrecherche (insbes. digitale Bibliotheken, Google Scholar, CiteSeer…)
- Basisliteratur
- Muster einer Überblicksarbeit (i. A. aus ACM Computing Surveys)
- Muster eines Eintrags in das Literatur-Wiki
- Muster der gutachterlichen Rückmeldungen zu einem realen Konferenzbeitrag
- Erfassungsschablone (Excel) für die numerische Beurteilung von Standardkriterien

Erfahrungen 2007

Im Sommersemester 2007 wurde das Seminar von 16 Studierenden belegt. Im Laufe des Semesters haben sich zwei der Studierenden wegen Überlastung abgemeldet, die verbleibenden 14 haben positiv abgeschlossen. Das Ende der Lehrveranstaltung bereits Mitte Juni erlaubt den Studierenden, sich zu Semesterende noch voll auf den Abschluss anderer Lehrveranstaltungen zu konzentrieren.

Klagenfurt, 18. 6. 2007 Martin Hitz, Rudolf Melcher

[3] Der Normalwert (= Note für eine ordentliche Leistung) liegt bei *Full Papers* bei 2, bei *Work in Progress Papers* bei 3. Besonders gute Gesamtleistungen können diesen Normalwert um einen Grad verbessern, genauso kann er um einen Grad verschlechtert werden, was ein Beurteilungsintervall von 1-3 bzw. 2-4 ergibt. Die Note 5 wird nur bei Nichterfüllung notwendiger Bedingungen vergeben.

Inhalt

Mobile Interfaces

Roman Metschina
Der Einzug von mikromechanischen Sensoren (MEMS) in elektronische Geräte 1

Wolfgang Rabl
Taktile Interfaces - Anwendung in mobilen Endgeräten 8

Daniel Felsing
Visual Mobile Interaction 18

Marcel Bricman
Navigation in großen Dokumenten auf kleinen Bildschirmen 30

Visualisierung

Alexander Abl
Using semantically enriched information and metaphors to visualize knowledge 38

Nadja Marko
Design von Werkzeugen für interaktive fortgeschrittene (2D) Informationsvisualisierungen 48

2D / 3D Basistechnologien

Stefan Urabl
3D-Dateiverwaltungssysteme 58

Christian Kruschitz
Fenstermanipulation in 2D Desktopumgebungen und Anwendbarkeit dieser Techniken an Touch Screens 66

Marlene Stroj
3D Desktop Effekte 76

Nicht-klassische Interfaces

Vera Koren
Brain-Computer Interface - Wo Mensch und Maschine aufeinander treffen 87

Martin Florian
Tangible Interfaces und Sensor-Network-Interface-Entwicklungen für den effizienten Datenzugriff 95

Ursula Dittrich
Interaktive Oberflächen – Verschiedene Technologien im Vergleich 102

Benutzerzentriertes Design

Gabriele Pibal
Interface Design digitaler Bibliotheken für Kinder 112

Der Einzug von mikromechanischen Sensoren (MEMS) in elektronische Geräte

Roman Metschina
Längdorf 63, 9184 St. Jakob i.R.
rmetschi@edu.uni-klu.ac.at
+436502802839

ABSTRAKT
Wer kennt es nicht – langweilige Spiele auf mobilen Geräten und Konsolen, Navigationsgeräte welche die Position in Stockwerken, Schluchten, Parkhäusern nicht ermitteln können, verwackelte Fotoaufnahmen mit Digitalkameras, Computer die bei starken Erschütterungen oder einem Sturz kaputt gehen oder mobile Geräte deren Displayhelligkeit sich nicht automatisch verändert. Alle diese Probleme, aber auch noch viele mehr, werden durch den Einzug von mikromechanischen Sensoren verbessert bzw. gelöst. Mikromechanische Sensoren – international auch unter dem Kürzel MEMS (Micro-Electro-Mechanical-Systems) bekannt, sind ein Ergebnis der Miniaturisierung in der Mechanik. MEMS-Sensoren sind nur wenige Quadratmillimeter groß, ihre mechanischen Strukturen haben Federn, Balken, Gewichte und Membranen mit Abmessungen in der Größenordnung von tausendstel Millimetern. Durch deren Einsatz ergeben sich eine Menge von neuartigen Interaktionsmöglichkeiten, welche den Umgang mit elektronischen Geräten vereinfachen sollen.

EINLEITUNG
Obwohl es keinem so richtig bekannt ist und wir es eventuell auch unbewusst ignorieren, hat der Einzug der MEMS in unser alltägliches Leben bereits seit Langem begonnen. Egal ob im Auto, beim Fotografieren eines Gebäudes mit der Digitalkamera oder beim Telefonieren mit einem Mobiltelefon, in vielen elektronischen Geräten sind MEMS bereits integriert. Diese Arbeit soll einen Überblick über die verschiedensten Arten von mikromechanischen Sensoren und deren Anwendungsgebiete geben. Des weitern wird auf die daraus neu entstandenen Interaktionsmöglichkeiten bei elektronischen Geräten eingegangen. In der Folge wird speziell das neue Apple iPhone unter die Lupe genommen, da dieses Gerät, welches erst im Herbst auf den europäischen Markt kommen wird, bereits eine Reihe dieser Sensoren integriert haben wird. Letztendlich wird ein Überblick über den Einzug der MEMS in die Spieleindustrie gegeben und es werden zukünftige Aussichten und Entwicklungen vorgestellt.

Autor Schlüsselwörter
MEMS, Mikromechatronik, Mikromaschinen, Mikrosystemtechnologie (MST), mikroelktromechanischen Systeme

Geschichtlicher Hintergrund
Einige Vorreiter der MEMS gab es schon in den 70er Jahren, wobei damals noch die Qualität eines Sensors bzw. Messsystems sehr stark vom handwerklichen Geschick des Herstellers abhang. z.B.: ein Sensorsystem mit einer Rohrfeder welche mit einem Potentiometer gekoppelt worden ist um den Druck zu messen. Mitte der siebziger Jahre wurden erstmalig Halbleiter als Sensoren eingesetzt, daraus resultierten dann die ersten Silizium Drucksensoren. Durch den technischen Fortschritt in der Mikroelektronik, wurde es bereits Ende der 70er und in den 80er Jahren ermöglicht, miniaturisierte Sensorsystme vor allem auf Festkörper und Silizumbasis herzustellen. Die großen amerikanischen, japanischen und europäischen Technologiekonzerne gewannen als Hersteller solcher miniaturisierten Sensorelemente an enormer Bedeutung.

Durch diese stürmische Entwicklung der Mikroelektronik wurden wichtige Grundlagen für die MEMS Technologie gelegt. Mit der Verbreitung der Halbleitertechnologien wurden die bis dahin in Mitteleuropa verbreiteten elektro-feinmechanischen Messysteme durch vollelektronische Systeme ersetzt bzw. verdrängt. So wurden für Druck- und Magnetfeldsensoren mit der Dünnschichttechnik die Prozesse zur Herstellung dünnster Membranen in Silizium entwickelt, welche heute in die dreidimensionale Formgebung der Mikromechatronik und MEMS Technologie überfließt. Heutzutage gibt es MEMS der zweiten Generation. Diese Sensorsysteme können Bewegungen, Geräusche, Lichtreize, Geruchsstoffe oder Farbveränderungen aufnehmen und sind teilweise sogar in der Lage, darauf in entsprechender Weise zu reagieren [MMS05].

Begriffsbestimmung

Bevor die einzelnen MEMS-Sensoren erläutert werden, möchte ich eine kurze Begriffsbestimmung mit ein paar MEMS Definitionen wiedergeben. Wie bereits im Abstrakt erwähnt steht MEMS für Micro Electro Mechanical Systems in der deutschen Sprache Mikroelektromechanische Sensorsysteme.

Unter dem Oberbegriff der "Mikroelektromechanischen Sensorysteme" (MEMS) wird die Integration von Sensoren und Aktuatoren der Mechatronik, der Mikrosystemtechnik der Mikrooptik, der Mikrofluidik und der Mikroelektronik zusammengefasst [MMS05].

Mikroelektronische Schaltungen können als Gehirn eines Systems gesehen werden und MEMS erweitern diese Entscheidungsfähigkeit mit Hilfe von Augen und Armen um einem Mikrosystem das fühlen und kontrollieren der Umwelt zu erlauben [MEM00].

Mems können definiert werden als eine Kombination aus Mikrosensoren und Mikroaktuatoren und elektronischen Bauelementen integriert auf einem einzelnen Chip [MAT].

Mikro-elektro-mechanische Systeme sind mikroskopisch-mechanische Systeme gekuppelt mit elektrisch oder elektronischen Schaltkreisen [DOA04].

Nach einigen Begriffsbestimmungen kann man sagen, dass MEMS sehr viel können aber auch sehr kompliziert sind. Zusammenfassend würde ich sagen das ein MEMS eine Vermischung aus verschiedensten Sensoren und Aktuatoren (Ein Aktuator wird in der Literatur definiert als ein Stellglied, das meistens elektrisch angesteuert wird und dessen Ausgangsgröße eine Energie oder Leistung ist [AKT07]) ist, welche auf einem wenige Quadratmillimeter großen Chip aufgebracht sind und in der Lage sind verschiedenste Wahrnehmungen wie Beschleunigung, Drehrate, Druck,… zu erfassen.

MEMS ersetzen Produkte, die bisher aus mehreren Bauteilen aufgebaut waren und vereinen so zwei oder mehr Funktionen der Mikroelektronik, Mikromechanik oder Mikrooptik. Als Basismaterial solcher kleinen Chips dient das Silizium, so wie in der gesamten Halbleiter- und Elektronikwelt. Durch die Siliziumtechnologie ist es möglich, kleine Sensoren von wenigen Quadratmillimetern Größe und mit hoher Zuverlässigkeit kostengünstig herzustellen.

Auf einem Funktionsträger – meist einem Silizumchip – gemeinsam integriert, stellen kleinste Sensoren, Aktuatoren, mechatronische Mikrosysteme und integrierte Schaltungen (sog. "ICs", Integrated Circuits oder "ASICs", Pllication Specific Integrated Circuits) eine ausgezeichnete Basis zur Lösung verschiedenster Problemstellungen dar [MMS05].

SENSORARTEN UND DEREN ANWENDUNGSGEBIETE

Durch die winizige Sensorik werden mechanische Messgrößen wie Druck, Beschleunigung, Drehrate, Pegelstand oder Massendurchfluss erfasst. Diese Sensoren wandeln physikalische oder chemische Messgrößen in elektrische Signale um und stellen diese Informationen elektronischen Systemen bereit. Ein Sensor wandelt also eine meist nichtelektrische Messgröße in ein elektrisches Ausgangssignal um. Sensoren werden häufig auch als Wandler bezeichnet. Da diese Umwandlung nicht immer ohne Fehler funktioniert, muss man auch mit Störgrößen wie z.B.: Temperaturschwankungen, schwankende Versorgungsspannungen rechnen.

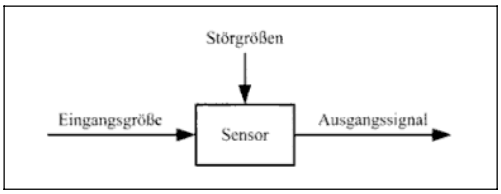


Abbildung 1. Sensor im Überblick [AUT06]

Bevor die einzelnen Sensorarten erwähnt werden, wird in Abbildung 2 eine Veranschaulichung über die Größe eines solchen MEMS gezeigt, verglichen mit einer Penny-Münze.

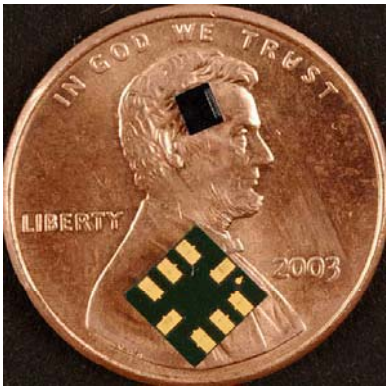


Abbildung 2. Spielekonsolen [SPK07]

Aufgrund der am Anfang erwähnten mechanischen Messgrößen kommt es zu folgender Einteilung der Sensoren [BOS07].

Inertialsensoren

Inertialsensoren beinhalten Beschleunigungs-, Drehraten- und Neigungssensoren. Der Begriff "Inertial" stammt aus der Physik und wird von dem Wort Inertialsystem abgeleitet. Ein Inertialsystem (von lateinisch iners „untätig, träge") ein Koordinatensystem, in dem sich jedes Objekt mit Masse, auf das keine Kraft wirkt, (bzw. die Summe aller Kräfte gleich Null ist), gleichförmig geradlinig bewegt

oder in Ruhe bleibt. Ein Inertialsystem ist also ein Bezugssystem in dem das Newtonsche Trägheitsgesetz uneingeschränkte Gültigkeit besitzt [TOM]. In Abbildung 3 sieht man einen Inertialsensor in vergrößterter Darstellung.

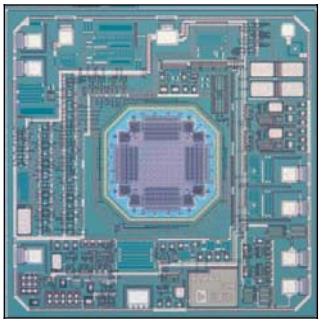


Abbildung 3. Inertialsensor [MEM07]

Beschleunigungssensoren

Mikromechanische Beschleunigungssensoren (vergrößerte Darstellung in Abbildung 4) im englischen auch Accelerometer genannt werden für die Ermittlung von Linear- oder Rotationsbeschleunigungskräften verwendet. Die Messgröße hat oftmals die Einheit der Erdgravitation (1g = ca. 9.81m/s2). Sofern man sich auf das Erdgravitationsfeld bezieht erhält man die Neigung bzw. der Neigungswinkel des Objektes [BES].

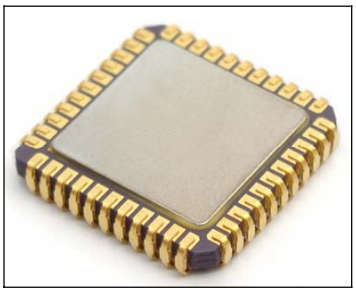


Abbildung 4. Beschleunigungssensor [IMI1]

Die Erfassung der Beschleunigung erfolgt mittels eines mikromechanischen Resonator, welche durch die Beschleunigung ausgelenkt wird. Dies wiederum kann durch ein geeignetes Messverfahren erfasst werden. Durch den Resonator kann eine Geschwindigkeitszunahme oder –abnahme bestimmt werden. [PRA] Verwendet man mehrere Sensoren in verschiedenen Winkelstellungen, lassen sich mehrdimensionale Bewegungsvorgänge ermitteln [BES].

In Abbildung 5 sieht man eine Skizze vom Aufbau eines Beschleunigungssensors und es folgt eine kurze Erklärung anhand eines Beispieles.

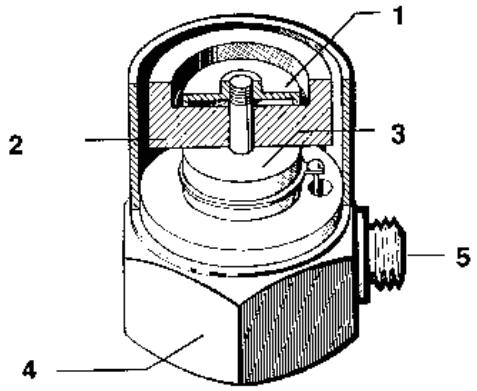


Abbildung 5. Beschleunigungssensor Skizze [BSS07]

Der Beschleunigungssensor (piezoelektrischer), es gibt eine Vielzahl anderer Beschleunigungssensoren besteht aus folgenden Elementen:

1. Vorspannfeder
2. seismische Masse
3. piezoelektrische Scheiben
4. Sockel
5. Anschlussbuchse

Um das Messprinzip zu verstehen, stellt man sich am Besten folgendes Szenario vor: Man sitzt in einem Auto auf dem Autositz und ist sozusagen die Vorspannfeder. Der Autositz ist die piezoelektrische Scheibe. Das Auto beschleunigt nun innerhalb von einer Sekunde von 0 auf 100 km/h. Was passiert? Sie werden von der Beschleunigungskraft in den Autositz gedrückt. In diesem Moment würde durch den Piezoeffekt (Schlag auf einen Kristall, die dann eine Spannung erzeugt, Beispiel Feuerzeug) eine Spannung zwischen Ihrem Körper und dem Autositz entstehen. Diese Spannung wird dann genommen und auf einem Computer umgerechnet, sodass man die Beschleunigung (Mass: m/s2) ablesen kann [BSS07].

Anwendungsgebiete:

Die Anwendungsgebiete für Beschleunigungssensoren sind sehr unterschiedlich und können somit in verschiedene Beschleunigungsbereiche (gering, mittel, hoch) eingeteilt werden [BES]:

- Stabilisierung und Lageregelung von Flugzeugen und Fahrzeugen
- Navigationssysteme
- Körperbewegungsanalyse
- Sensorik zur Auslösung von Airbags
- Aktive Federungssysteme / Fahrdynamiksysteme für Kraftfahrzeuge

- Schwingungsanalyse (z.B. von Gebäuden oder Maschinen)
- Fallschutz (z.B. in Festplatten von Laptops)
- Alarmanlagen bei beweglichen Gütern [BES]
- Steuerung von Videospielen (Wii) [PRA]
- Seismik und Erdbebenvorhersage [BES]

Drehratensensoren
Drehratensensoren welche in der Abbildung 6 zu sehen sind, sind auch unter dem Namen Gyroskope bekannt und messen die Rotationsgeschwindigkeit eines Körpers. Wie im Namen bereits versteckt dienen sie allgemein zur Erfassung der Drehgeschwindigkeit bzw. Drehrate eines Objektes [AUT06].

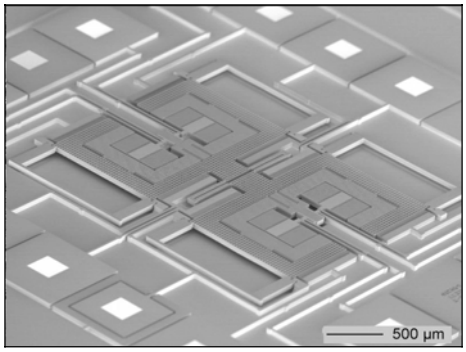


Abbildung 6. Vergrößerung Drehratensensor [DRE07]

Als Messprinizp verwenden sie die Wirkung des Sagnac-Effektes in [SAG02] oder der Coriolis-Kraft in [COR01].

Anwendungsgebiet

- Bildstabilisierung in Digitalkameras, Mobiltelefonen, PDA,…
- Sensorik für Plattformstabilisierung (Modellflugzeuge und Hubschrauber,…)
- Spielekonsolen für neuartige Interaktionsmöglichkeiten (Wii, PS3)
- Fahrzeug-Navigationssysteme mit Koppelnavigations-Fähigkeit (Zusammenführung von Informationen aus mehreren Sensoren) (PDF-Drehratensensoren-Deutsch)
- Anzeigen eines Überschlafunfalls bei einem PKW (Mikrosystemtechnik.: Konzepte und Anwendungen. Von Ulrich Mescheder)
- Steuerung im Bereich virtueller Realität (3D-Computermaus) (Mikrosystemtechnik.: Konzepte und Anwendungen. Von Ulrich Mescheder)

Neigungssensoren
Neigungssensoren messen die Abweichung eines Körpers gegenüber der Horizontalen. Da diese Messung im Prinzip auf die Messung des Erdbeschleunigungsvektors zurückführt, handelt es sich auch oft um modifizierte Beschleunigungssensoren [NEI02].

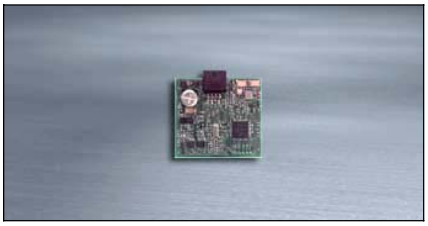


Abbildung 7. Neigungssensor [NEG07]

Anwendungsgebiete

- Leistungsabhängige Steuerung von Automatikgetrieben
- Diebstahlsanzeige von Fahrzeugen
- Elektronische Wasserwagen
- Dreidimensionale Bilder auf PDAs – es wird die horizontale Kippbewegung eines PDAs ermittelt, je nach dem auf welche Seite man den PDA kippt, wird eine andere Ansicht eines gerade angezeigten Bildes angezeigt. [NEI02]

Drucksensoren
Der Druck wird in der Physik als “Kraft pro Fläche” definiert. Der Druck A ist also das Ergebnis aus der Kraft F durch die Fläche A (p=F/A). Mit diesen Grundinformationen wird deutlich, wie Druck gemessen werden kann. Wird ein Sensor in ein Medium gebracht, dann wirkt eine Kraft senkrecht zur Sensorfläche. Wenn diese Kraft zu einer mechanischen Verformung führt, kann also im Grunde über eine Wegmessung der Druck bestimmt werden. Bei elektronischen Geräten wird die Längenänderung in ein elektrisches Signal umgesetzt. Abbildung 8 veranschaulicht einen Kolben, der annähernd reibungslos verschoben werden kann. Auf der einen Seite herrscht der zu messende Druck, auf der anderen Seite ist eine Feder angebracht. In Abhängigkeit vom Druck wird die Feder mehr oder weniger komprimiert. Aus der Länge der Feder lässt sich der Druck bestimmen [IFM07]. Eine wichtige Anforderung an Drucksensoren ist, dass sie gegen bestimmte Medien beständig sein müssen. So muss z.B.: ein Drucksensor für das Bremssystem gegen den Kontakt mit Bremsflüssigkeit beständig sein [HEI].

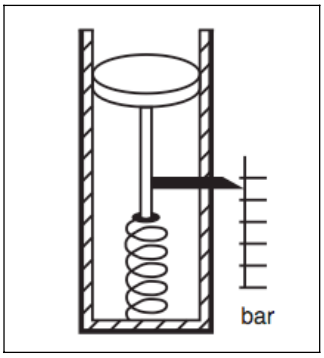


Abbildung 8. Drucksensor Prinzip [IFM07]

Es gibt eine Vielzahl an unterschiedlichen Drucksensoren auf dem Markt, so z.B.:

- Passivdrucksensoren
- Relativdrucksensoren
- Absolutdrucksensoren
- Differenzdrucksensoren

Auf diese Unterarten von Drucksensoren möchte ich nicht näher im Detail eingehen.

Anwendungsgebiete

- Automobilindustrie (Kraftstoffdruck, Bremsdruck, Reifendruck,…) [HEI]

MEMS IN DER SPIELEINDUSTRIE

MEMS sind in vielen Bereichen wie der Automobilindustrie, Medizin, Sport, Sicherheitstechnik bereits vertreten. Der Einzug in die Spieleindustrie hat aber erst vor kurzer Zeit begonnnen. So müssen die Fragen gestellt werden:

- "Welche neuen Möglichkeiten bieten die MEMS in der Spieleindustrie?
- "Was sind die daraus resultierenden Interaktionsmöglichkeiten?"
- "Werden diese von den Spielern akzeptiert?"

Um die Fragen zu Beantworten, wird ein Vergleich der zur Zeit am Markt größten und beliebtesten Spielekonsolen und zwar der Playstation 3 von Sony, der XBOX 360 von Microsoft und der Wii von Nintendo vorgenommen.
Die Unterscheidungskriterien beziehen sich auf die verwendeten Sensoren und auf die sich dadurch ergebenen Interaktionsmöglichkeiten.

Abbildung 9. Spielekonsolen [SPK07]

Interaktion mit der Wii

Schon in der Bedienungsanleitung wird darauf hingewiesen, dass die Wii-Controller (Nunchuk und Wii Remote – siehe Abbildung) nicht zu heftig geschwenkt werden können, um mögliche Mitspieler nicht zu verletzen.
Bei Wii Sports wird der Controller zum virtuellen Golfschläger und man kann ihn auch als Baseball- oder Tennisschläger verwenden.

Die Interaktion erfolgt mittels zwei Controllern (Nunchuck und WiiMote) welche miteinander verbunden werden können. Die Sensoren für die Wii wurden von der Firma STMicroelectronics NV in der Schweiz hergestellt. Es handelt sich hierbei um Beschleunigungssensoren welche in beiden Controller integriert sind [TIN07]. Daraus ergeben sich folgende Interaktionsmöglichkeiten:

- Die WiiMote fungiert als Angelrute und der Nunchuck als Kurbel.
- Die WiiMote fungiert als Schwert bzw. Schild und man kann ihn horizontal und vertikal schwingen.
- Mit dem Nunchuck kann man ein Snowboard lenken und die WiiMote dient zum Springen.
- Sportspiele: Boxen, Bowlen, Tennis spielen, Golfen…alle diese Sportarten werden mit den gleichen Bewegungen wie in Echt ausgeübt.

iPHONE

Das iPhone, welches in der Abbildung 5 zu sehen ist, ist das neue Flaggschiff der Firma Apple. Es ist eine Kombination aus einem Mobiltelefon, einen iPod mit berührungssensitivem Breitbild-Display und ein Internet-Kommunikationsgerät mit E-Mail, Webbrowser,... [APP07]

Abbildung 11. Apple iPhone [IPH07]

Unterschied zu herkömmlichen Smartphones

Smartphones sind nicht so smart und auch nicht so leicht zu bedienen wie es Ihr Name verspricht. Sie besitzen Plastiktastaturen, sind des Öfteren kompliziert und schwierig zu benutzen. Die Steuerungs-Buttons sind in der Oberfläche fixiert, können nicht verändert werden und verschwenden außerdem noch Platz. Im Vergleich zum Computer mit der Maus – wurde das Problem gelöst, doch wie soll die Interaktionsmöglichkeit zwischen Mensch und PDA (iPhone) funktionieren. Steve Jobs gab die Antwort auf der letzten Apple Keynote bekannt: mit dem Finger. Steve Jobs: *„Wir sind alle mit dem ultimativen Zeigegerät geboren worden - unseren Fingern - und iPhone nutzt sie, um die revolutionärste Benutzeroberfläche seit der Maus zu schaffen."* Die Bedienung des Apple iPhones erfolgt also mittels Finger.

Das iPhone wird in den USA am 29. Juni 2007, in Europa Ende 2007 und in Asien 2008 auf den Markt kommen. Da diese Arbeit auf MEMS basiert, möchte ich nicht näher auf die Details wie Kamera, Touchscreen, Akku,...des iPhones eingehen sondern mich mehr auf die Sensoren konzentrieren, welche das iPhone integriert haben wird.

Sensoren des iPhones

Das iPhone verfügt über drei hochentwickelte eingebaute Sensoren. Diese Sensoren sind ein Beschleunigungssensor, ein Annäherungssensor und ein Umgebungslichtsensor - die automatisch für erhöhten Bedienungskomfort und eine längere Batterielaufzeit sorgen [MAC07].

Der Beschleunigungssensor merkt, wenn der Anwender das Gerät vom Hoch- ins Querformat gedreht hat und richtet den Inhalt entsprechend auf dem Display aus. Damit lässt sich etwa eine Website in der ganzen Breite oder ein Foto bildschirmfüllend im Querformat wiedergeben [APP07].

Der eingebaute Annäherungssensor registriert, sobald das iPhone ans Ohr gehoben wird und schaltet das Display ab, um Energie zu sparen und versehentliche Berührungen solange auszuschließen, bis das iPhone wieder wegbewegt wird [IMI].

Der Umgebungslichtsensor passt die Helligkeit des Displays automatisch an die gegebenen Lichtverhältnisse an, um die Darstellung zu verbessern und Energie zu sparen [APP07].

Interaktionsmöglichkeiten durch das iPhone

Durch die eingebauten Sensoren ergeben sich folgende neuartigen Interaktionsmöglichkeiten:

- Um einen Ausschnitt einer Website oder ein Foto zu vergrößern, muss man mit zwei Fingerspitzen auf das Display tippen und die Finger auseinanderziehen [APP07]
- Listen durchblättern per Fingerspitze (Tap) [APP07]
- Zum Entsperren des iPhones dient einfach ein ziehen des Fingers von der linken zur rechten Seite des Touchscreens

ZUKUNFTSAUSSICHTEN UND RESÜMEE

Angefangen hat alles in den späten 80er Jahren mit den Einzug von mikroelektromechanischen Systemen in die Automobilindustrie (Aktivierung der Airbags, Bremsdruck,…), Luftfahrt, Armee. Jetzt ist es so, dass diese jahrelangen Erfahrungen in den verschiedensten Branchen in die Konsumerelektronik übernommen werden. Vorraussetzung dafür sind rapide sinkenden Chippreise, welche noch vor nicht allzu langer Zeit extrem teuer waren (z.B.: ein Chip welcher noch vor einigen Jahren für die Steuerung einer Flugrakete zuständig war, kostet heute nicht mehr $25.000 sondern $2) [TIN07]. Dieser Preisabfall sorgt dafür, das die MEMS ihren weg fortsetzen können. Doch wohin geht ihr Weg? Wie bereits erwähnt gibt es schon einige elektronische Geräte wie das iPhone, Nokia 5500 (mit eingebautem Beschleunigungssensor für Spiele und zum Steuern des Musikplayers) [RHKC07], Laptops welche durch einen eventuellen Aufprall ihre Festplatte schützen, Digitalkameras die verwackelte Aufnahmen wieder begradigen usw. Dies sind nur ein paar Beispiele aus einem Bereich, doch es gibt noch einige andere Gebiete welche bald von MEMS erobert werden wie z.B.: das Erforschen der Flugdynamik von Insekten mit Hilfe von MEMS-Kraftsensoren, die Erforschung des menschlichen Körpers oder Sensoren welche in Kleidung integriert werden um so den Träger bezüglich der Haltung, Bewegung informieren zu können – sensible Kleidung [BUL07]. Letztendlich lässt sich sagen, dass die kleinen winzigen mikroelktromechanischen Systeme enorm große Auswirkungen auf die elektronischen Geräte, welche wir in unserem alltäglichen Leben gebrauchen, haben bzw. haben werden.

REFERENZEN

[RHKC07] Ronkainen S., Häkkilä J., Kaleva S., Colley A., Linjama J.: Tap Input as an Embedded Interaction Method for Mobile Devices. In: Proceedings of the 1st international conference on Tangible and embedded interaction TEI '07

[TIN07] Svensson P.: Tiny springs let Wii and PlayStation 3 controllers sense motion. URL:http://www.usatoday.com/tech/gaming/2006-11-20-console-controllers_x.htm, Downloaddatum: 10.06.07

[DOA04] Mahmoudian N., Nakhaie Jazar G., Rastgaar Aagaah M., Mahinfalah M.: Dynamics of a Micro Electro Mechanical System (MEMS). In: Proceedings of the 2004 International Conference on MEMS, NANO and Smart Systems (ICMENS'04)

[BOS07] Bosch: Presseinformation: Mit 100 Millionen mikromechanischen Sensoren pro Jahr auf Spitzenniveau. URL:http://www.bosch-presse.de/TBWebDB/de-DE/PressText.cfm?id=2567, Downloaddatum: 18.4.07

[MIK] Mescheder U.: Mikrosystemtechnik: Konzepte und Anwendungen. S. 61 und folgende

[AUT06] Reif K.: Automobilelektronik. Eine Einführung für Ingenieure, 05/2006

[COR01] Bergmann L., Schaefer C., Raith W.: Lehrbuch der Experimentalphysik: Erde und Planeten. Gruyter, 2. Auflage. April 2001

[AUT07] AUTOMOBIL ELEKTRONIK - 02/2007; MEMS-Sensoren im Überblick.

Internetadressen

[BUL07] ETH Zürich: Mikrosysteme – Nanosysteme. Bulletin Nr. 292 Februar 2004. URL:http://www.cc.ethz.ch/news/bulletin/pdf/eth_bulletin_292.pdf, Downloaddatum: 12.06.07

[MEM00] Memsnet. URL: www.memsnet.org/mems/what-is.html, Downloaddatum: 30.05.07

[MMS05] Glück M.: MEMS in der Mikrosystemtechnik (Broschiert - Juli 2005)

[TOM07] Der iMacPhonePod. In: Tommorrow – Enjoy digital life!, März 2007, S. 56 – 59

[MAT] MEMS World: Design, fabrication, applications. URL: http://matthieu.lagouge.free.fr/mems/, Downloaddatum: 19.5.07

[APP07] Presseinformation Apple. URL: http://www.apple.com/de/pr/pr-infos2007/januar/iphone.html, Downloaddatum:

[WIK] Wikipedia Drucksensor. URL: http://de.wikipedia.org/wiki/Drucksensor, Downloaddatum:

[HSG] URL: http://www.hsg-imit.de/index.php?id=40, Downloaddatum: 30.05.07

[BES] Beschleunigungssensoren. URL: http://de.wikipedia.org/wiki/Beschleunigungssensor, Downloaddatum: 02.06.07

[PRA] URL: http://www.praktika.de/praktikum/praktikanten/stellen.17979.html, Downloaddatum: 02.06.07

[TOM] URL: http://www.tomorrow.de/news?id=360046, Downloaddatum: 02.06.07

[APP07] Apple drängt ins Handy-Geschäft und in die Wohnzimmer. In: c't 3/2007, S. 18, URL: http://www.heise.de/ct/07/03/018/, Downloaddatum: 18.5.07

[MAC07] Macnotes. URL: http://www.macnotes.de/2007/01/11/iphone-im-uberblick/, Downloaddatum: 31.05.07

[IMI] URL: http://www.hsg-imit.de/index.php?id=41&L=0, Downloaddatum: 03.06.07

[IMI1] URL: http://www.hsg-imit.de/index.php?id=40, Downloaddatum: 03.06.07

[HEI]URL: http://www.heise.de/newsticker/meldung/26816

[BSS07] Bild http://www.it-academy.cc/article/1116/Der+Beschleunigungssensor.html

[SAG02] Gevatter HJ.: Automatisierungstechnik 1. Meß- und Regeltechnik. Springer, Berlin. 1. Auflage. Mai 2000

[IPH07] URL: http://www.apple.com/pr/products/iphone/iphone.html, Downloaddatum: 15.5.07

[DRE07] URL: http://idw.tu-clausthal.de/pages/de/news190887, Downloaddatum: 20.5.07

[AKT07] Aktuator – Aktor. URL: http://www.ipi.uni-hannover.de/html/lehre/diplomarbeiten/1999/geisler.jens/aktoren.htm

[SPK07] http://www.chip.de/bildergalerie/c1_bildergalerie_v1_25181777.html?show=4

[IFM07] IFM Electronic: Schulungsunterlagen – Drucksensoren, Stand März 2003, URL: weblx.homelinux.net/jsp/Sensoren/Schulungsunterlagen/S500d.pdf

[NEI02] Mit Neigungssensoren zum dreidimensionalen PDA-Display, URL: http://www.heise.de/newsticker/meldung/26816, Downloaddatum: 10.5.07

[MEM07] URL: http://matthieu.lagouge.free.fr/mems/hist_app.html, Downloaddatum: 20.5.07

[NEG07] Neigungssensoren. URL: http://rb-k.bosch.de/de/einsatzgebiete/verbrauch/vernetzung/fahrzeugsicherung/neigungssensoren.html.

Taktile Interfaces
Anwendung in mobilen Endgeräten

Rabl Wolfgang

wrabl@edu.uni-klu.ac.at
Mat.Nr.: 0161121

ABSTRACT
Der Bereich der mobilen Endgeräte charakterisiert sich insbesondere durch deren beschränkte Interaktionsmöglichkeit mit dem Benutzer. Der Informationsfluss zum Benutzer wird auf Grund der geringen Baugröße und der somit recht kleinen vorhandenen Displays erheblich eingeschränkt. Des Weiteren gestalten sich auch die Eingabemöglichkeiten schwieriger im Vergleich zu den üblichen Benutzerschnittstellen.

Diese Arbeit möchte sich der Thematik des ersteren Problems näher annehmen und wird aufzeigen, wie sich die Benutzerinteraktion durch Hinzufügen einer weiteren Dimension verbessern kann. Hierbei handelt es sich um die Übermittlung taktiler Information zusätzlich zu bereits vorhandenen audiovisuellen Eindrücken. Der Benutzer bekommt also während seiner Interaktion haptisches Feedback verschiedenster Art. Hierbei kann es sich um Druck, Vibrationen, Temperaturveränderungen usw. handeln.

Es gibt in diesem Bereich bereits Unmengen an viel versprechenden Arbeiten und sehr viele lauffähige Prototypen, jedoch sind viele dieser Technologien gar nicht oder nur sehr begrenzt im mobilen Bereich einsetzbar. Aus diesem Grund beschäftigt sich ein Teil der Arbeit damit, taktile Schnittstellen zu finden, welche in portablen Geräten Verwendung finden können.

Weiters wird darauf eingegangen, wie sich die Interaktion aufgrund der Verwendung solcher neuen Interfaces ändern und verbessern kann. Interessant ist in diesem Zusammenhang auch die Frage, wie sich eine mögliche Kombination mehrerer unterschiedlicher Technologien hier auswirken könnte und in welchen Bereichen offene Fragen eine weitere Forschungstätigkeit erlauben bzw. notwendig machen.

1.EINLEITUNG

Spricht man in der Praxis von „Visualisierung“, so stellt man sich im Allgemeinen den Prozess der graphischen Darstellung diverser Datenmengen vor. Dies mag natürlich richtig sein, jedoch wird damit nicht der gesamte Bereich umschrieben. Abgesehen von den optischen Eindrücken, nimmt der Mensch Informationen aus der Außenwelt auch mithilfe seiner weiteren Sinne auf. Aus diesem Grund muss man weitergehen und den Prozess des Visualisierens dadurch definieren, dass man etwas unsichtbares, abstraktes, unangreifbares in eine Form überführt, die der Mensch wahrnehmen kann. Neben sehen und hören bietet sich hier auch Berührung und ansprechen des Tastsinnes an. Diese Art der Wahrnehmung wird auch als haptische Wahrnehmung bezeichnet. Man nimmt also Informationen aus der Umgebung auf, in dem man sie berührt, dabei Widerstände spürt und Oberflächenmuster erkennt.[1]

Genauer definiert sich der Begriff Haptic wie folgt:

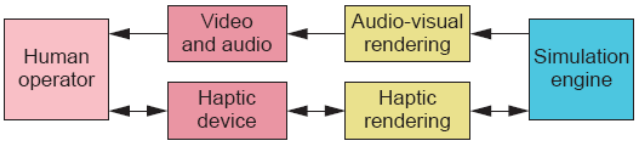
„Als haptische Wahrnehmung (griech.: haptikos = greifbar, umgangssprachlich auch Tastsinn) bezeichnet man eine Sinneswahrnehmung von Lebewesen, mit der bestimmte mechanische Reize wahrgenommen werden können. Die Gesamtheit der haptischen Wahrnehmungen erlaubt es dem Gehirn, Berührungen, Druck und Temperaturen zu lokalisieren und zu bewerten. Es wird unterschieden zwischen taktiler Wahrnehmung (Oberflächensensibilität) und kinästhetischer Wahrnehmung (Tiefensensibilität).“[2]

Der erwähnte kinästhetische Teil bezieht sich auf Sinneseindrücke, welche sich aus der räumlichen Lage des Körpers und der Spannungszustände der Muskeln ergeben, soll uns aber im weiteren Verlauf der Arbeit nicht weiter interessieren, vielmehr wird der Fokus auf der taktilen Wahrnehmung zu liegen kommen. Hier wird über ein entsprechendes Interface Information in Form von tastbaren Texturen, Oberflächenrauheit und Temperatur übertragen.

Es gibt bereits sehr viele Forschungsgruppen auf diesem Gebiet und die meisten beschäftigen sich mit Anwendungen in virtuellen Realitäten oder bei Teleoperationen. Ein weiterer viel versprechender Bereich eröffnet sich aber erst bei der Integrierung solcher Interfacemodelle in kleine mobile Endgeräte. Hier will der Benutzer selbstverständlich keine großen plumpen Systeme mit sich führen, weshalb sehr viele bestehende Prototypen einfach Auf Grund ihrer technischen Ausprägung nicht eingesetzt werden können.

Neben der technischen Ausführung solcher Interfaces stellt aber auch die korrekte Überführung der Daten in die Entsprechenden haptischen Impulse einen wichtigen Aspekt

dar. Wie auch bei konventionellen grafischen Benutzerschnittstellen müssen die Informationen richtig gerendert werden, um für Benutzer verständliche Informationen zu liefern, wie hier in Abbildung 1 zu sehen ist.[3]



[3]Abbildung 1: Haptisches Rendern

Auch hier gibt es starke Abweichungen zu herkömmlichen Systemen, will man sich in den mobilen Bereich vor wagen. Hält der Benutzer ein Gerät in der Hand, ist es nicht möglich ein korrektes kinästhetisches Feedback zu liefern, da mithilfe heutiger Technologien keine Kraft auf den Benutzer gewirkt werden kann, die dazu führt, dass das mobile Gerät und somit die Hand des Benutzers die Lage im Raum ändert. Somit verliert man diese Dimension beim rendern und kann sich nur über taktile Möglichkeiten mitteilen.

2.MOBILE TAKTILE INTERFACESYSTEME

Was für Gründe gibt es, taktile Interfaces in den mobilen Bereich überzuführen? Die Möglichkeiten des Benutzers mit den Geräten zu interagieren sind hier sehr eingeschränkt. Es sind nur recht kleine Displays verfügbar, die Umgebungsgeräusche sind oft sehr laut, bzw. erlaubt die Umgebung keine Audio Ausgabe was z.B. in einer Bibliothek der Fall sein könnte. Hier kann man dadurch natürlich Abhilfe schaffen. Des Weiteren hat man nicht immer die volle Aufmerksamkeit des Benutzers, da dieser sich ja auch bewegt und auf sein Umfeld acht geben muss.

Aufgrund dieser Äußeren Einflüsse ist die Aufnahmefähigkeit über die regulären audiovisuellen Kanäle recht schnell erschöpft. Dem Gegenüber steht aber die zumeist ungenutze Aufnahmekapazität über haptische Impulse. Zudem kommt noch die Tatsche, dass der Mensch über den Tastsinn Informationen bis zu 5-mal schneller verarbeiten kann als das bei visuellen Eindrücken der Fall ist. [10] Ein nicht unerheblicher weiterer positiver Aspekt zeigt sich auch dadurch auf, dass der Benutzer eines mobilen Gerätes dasselbe die meiste Zeit in der Hand hält und somit ein direkter Kontakt jedenfalls gegeben ist.

Genau aus Diesen Gründen ist die Einbeziehung des Tastsinnes gerade im Interfacedesign mobiler Geräte von enormer Bedeutung und kann sehr viel zur Verbesserung der Interaktion beitragen, da mehr Informationen als bisher quasi „nebenbei“ mit übertragen werden können.[4]

3.FORSCHUNGSBEREICHE

Der nun folgende Teil dieser Arbeit befasst sich mit den verschiedenen Forschungsergebnissen und Prototypen von taktilen Interfaces im mobilen Bereich.

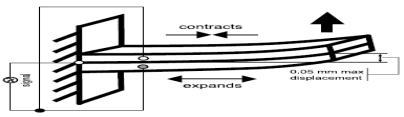
Es existiert bereits ein System, welches taktile Informationen über mobile Geräte übermittelt und heutzutage schon sehr weit verbreitet und in fast jedem Mobiltelefon oder ähnlichem integriert ist: der Vibracall. Natürlich handelt es sich hier nur um ein sehr einfaches Interface, dass nur einfache Information übermitteln kann. Jedoch gibt es Möglichkeiten zur Weiterentwicklung, was im folgenden Forschungsprojekt auch getan wurde.

3.1.Vibration

Während in herkömmlichen Mobiltelefonen nur sehr einfache Vibrationsmuster Verwendung finden, ist der Mensch in der Lage sehr viel mehr Information über diesen Kanal zu empfangen. Taktile Impulse können schnell hintereinander, im Bereich von 5ms, erfolgen, und man ist dennoch in der Lage das zu erkennen. Weiters kann man auch die Intensität und die Frequenz der Stimulation gut erkennen und unterscheiden. Zu guter letzt ist man auch in der Lage, verschiedene Wellenformen voneinander zu unterscheiden.

Führt man sich nun aber die Technologien vor Auge, welche in den heutigen Mobiltelefonen zur Anwendung kommen, um dem Anwender über Vibration Informationen zu übermitteln, kommt man schnell zu dem Schluss, dass diese nicht ausreichen werden, um komplexere taktile Muster zu übertragen. Die Motoren, welche verwendet werden haben zu lange Ansprechzeiten, als das man die erzeugte Vibration genau genug steuern könnte. Man ist auch nicht in der Lage, mehrere verschiedene Vibrationsmuster zu generieren, da diese durch die technischen Gegebenheiten des Motors und der Unwucht vorgegeben sind.

Hierzu war es also nötig, neue Aktoren zu entwickeln, welche dieser Anforderung gerecht werden konnten. Im konkreten Forschungsprojekt wurde der Motor durch zwei dünne piezokeramische Schichten ersetzt, zwischen denen sich Elektroden befanden. Je nach der angelegten Spannung zieht sich eine Schicht zusammen und die andere dehnt sich aus, somit verformt sich die Anordnung und verbiegt sich in die eine oder andere Richtung (Abbildung 2). Der entscheidende Schritt hierbei war die Verwendung vieler solcher extrem dünner Einheiten, anstatt einer größeren, wodurch die notwendige Spannung extrem verringert werden konnte, was einen Einsatz in mobilen Geräten überhaupt erst möglich machte. Nun ist es also möglich, Aktoren in verschiedensten Größen und Formen zu produzieren und somit auch beliebig in mobilen Geräten zu platzieren. Weiters können beliebige Vibrationsmuster generiert und zum Benutzer übermittelt werden.



[4]Abbildung 2: Piezokeramischer Aktuator

Man ist also sehr viel freier in der Gestaltung des gelieferten Feedbacks. [4]

Ausgehend von dieser Entwicklung ergeben sich nun Unmengen an Anwendungsmöglichkeiten. Bis zu diesem Zeitpunkt konnte über Vibration nur ein einziges Bit übertragen werden. Dem Benutzer wurde mitgeteilt, dass etwas geschehen ist. Um jedoch feststellen zu können worum es sich handelt, war es notwendig, seine aktuelle Handlung zu unterbrechen und sich ganz dem mobilen Gerät zu widmen. Mit Hilfe dieser nun vielfältigeren Vibrationsmuster kann der Benutzer über gewisse Tatsachen informiert werden, ohne seine aktuelle Tätigkeit unterbrechen zu müssen. So kann bei einem Anruf durch das taktile Interface beispielsweise auch die Identität des Anrufers oder die Dringlichkeit übermittelt werden. Aufgrund dieser Erkenntnis kann der Anwender entscheiden, ob er den Anruf annehmen möchte oder nicht ohne sich vom Umfeld abwenden zu müssen.

3.1.1.Taktile Fortschrittsbalken

Zwei weitere Anwendungsmöglichkeiten sind taktile Fortschrittsbalken und Feedback beim scrollen durch Anwendungen. Muss man in irgendeiner Anwendung auf die Fertigstellung warten, so wird das in visueller Form häufig als Fortschrittsbalken abgebildet. Damit man aber über den momentanen Stand im Bilde ist, muss man den Blick auf den Balken werfen. Und das jedes Mal aufs Neue. In der Zeit dazwischen hat man keinerlei Informationen über den aktuellen Status. Übermittelt man nun diese Daten über eine taktile Schnittstelle, muss man lediglich einen Kontakt zum Gerät haben und kann sich ansonsten auf andere Dinge konzentrieren. Der aktuelle Stand wird dadurch abgebildet, indem das Gerät in Abständen vibriert. Diese Abstände werden zunehmend kleiner, je näher sich der Prozess an der Fertigstellung befindet.[4]

Natürlich handelt es sich hierbei um eine recht simple Art der Codierung und es würden sich sicherlich noch interessante Forschungsmöglichkeiten ergeben auf welche Art und Weise man zusätzliche Möglichkeiten dieser Schnittstellenform wie Frequenz, Stärke oder Wellenformen verwenden könnte um solche Fortschrittsbalken noch zu erweitern oder besser darzustellen. Alleine ein simples Hinzufügen von unterschiedlichen Intensitäten könnte die Wahrnehmung eventuell schon verbessern, aber hierzu fehlen noch weitere Ergebnisse und Studien.

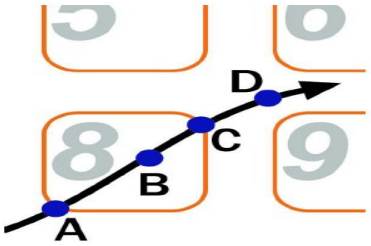
3.1.2. Unterstützung bei Gestensteuerung

Verwendet man tangible Interfaces um in Anwendungen je nach Bewegung des Gerätes zu scrollen, so fehlt es einem an entsprechendem Feedback. Man fährt oft über das Ziel hinaus oder stoppt zu früh. Integriert man hier ein taktiles Interface, welches entsprechend der Scrollgeschwindigkeit vibriert, so kann man die Benutzerfreundlichkeit erhöhen. Ein entsprechendes Experiment zeigte auf, dass durch die zusätzliche Information die Geschwindigkeit beim scrollen um 20% erhöht werden konnte, sowie die Fehlerrate bzw. das Verfehlen des gewünschten Zieles ebenfalls um 20% verbessert wurde. Man befindet sich hier also eindeutig auf dem richtigen Weg.

Dennoch, auch hier bietet sich noch sehr viel Spielraum für weitere Forschungsansätze. Inwiefern kann man verschiedene Vibrationsmuster einsetzen um die Eindrücke während des Bedienvorgangs anzureichern? Es wäre sicherlich auch möglich weiterführende Informationen zu übermitteln, beispielsweise durch Ändern des Feedbackmusters im Falle eines Links über den man scrollt und ein wieder anderes Muster bei Bildern oder ähnlichem.

3.1.3. Vibrierende Displays

Durch die Kombination dieser Schnittstelle mit visuellen Displays und Touchscreens kann überdies eine weitere viel versprechende Symbiose stattfinden. Werden auf ebensolchen Touchscreens Buttons oder ähnliche Eingabeelemente dargestellt, so führt das für den Anwender unweigerlich zu dem Problem, sich voll auf den dargestellten Inhalt fokussieren zu müssen. Es ist beispielsweise nicht möglich auf einem virtuellen Nummernfeld eine Telefonnummer zu wählen ohne hinzusehen. Schlicht aufgrund der Tatsache, dass man im Gegensatz zu einem herkömmlichen Display die Tasten nicht spürt. Nun, es gibt eine Reihe interessanter Ansätze, welche im späteren Verlauf der Arbeit aufgegriffen werden, jedoch lässt sich das eben genannte Problem auch recht einfach mittels folgender Methode entschärfen.



[5]Abbildung 3: Feedbackbereiche bei Benutzereingaben

Man rüstet ein Mobiles Gerät mit Aktoren aus, welche den gesamten Touchscreen zum Schwingen bringen. Fährt der Benutzer nun dieses Display entlang und kommt auf seinem Weg über einen Button, so kann eine Vibration den Übergang auf das Eingabeelement hinauf signalisieren. Verlässt man dieses wieder, so teilt das einem ein erneutes Vibrieren mit. Verweilt man auf dem Button, so bekommt man ein leichtes Schwingen zu spüren. Eines, welches mit der Zeit immer stärker wird was nach einer definierten Dauer dazu führt, dass man die Eingabe tätigt, den Button also gedrückt hat (Abbildung 3).

Noch besser kann dieser Vorgang natürlich mit Displays umgesetzt werden, welche auch die Stärke des Fingerdrucks ermitteln können. Hier kann das Verhalten sehr schön eins zu eins umgesetzt werden. Der Anwender fährt über das Display, bekommt durch die taktile Schnittstelle mitgeteilt, sich über einem Button zu befinden und kann durch

stärkeres drücken die Eingabe tätigen. Das alles wie gesagt natürlich ohne hinzu sehen.[5]

Mit dieser relativ einfachen Umsetzung kann man sehr schön das Gefühl vermitteln, mit realen Tasten zu arbeiten. Auch hier bietet sich vor allem wieder die Einbeziehung verschiedener Wellenformen an, da mit Hilfe dieser Differenzierung sehr leicht unterschiedliche Eingabeelemente identifiziert werden können. Fragen wie: „Handelt es sich hier nun um einen Button oder einen Scrollbalken?“ können damit sofort und sehr subtil beantwortet werden. Was in diesem Kontext jedoch leider Fehlt, bzw. nicht gefunden werden konnte sind ermittelte Resultate von Testpersonen, inwieweit sich die Eingabe durch Vibrationen von Touchscreens verbessern lassen kann.

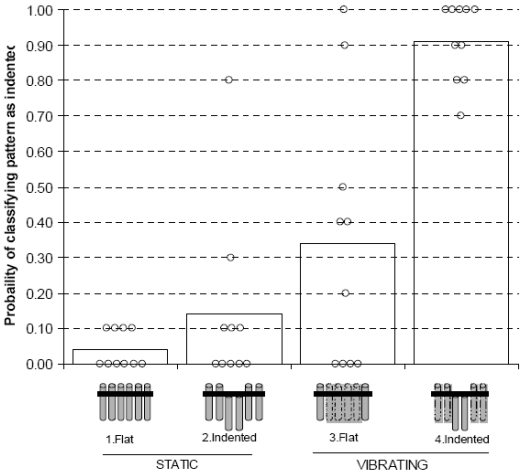
Wenn sich hier allerdings in zukünftigen Experimenten gute Resultate ergeben, so scheint mit dieser Methode eine einfache Umsetzung im Endbenutzerbereich möglich zu sein.

3.1.4. Rendern von Oberflächen

Mit dieser Form von Aktoren ist es allerdings nicht möglich statische Auslenkungen in ausreichender Form umzusetzen. Damit ist gemeint, dass das taktile Interfaceelement sich in eine Richtung ausdehnt und dann in dieser Position verbleibt. Die entsprechende Auslenkung ist dafür zu gering. Mit einer entsprechenden Anordnung mehrerer solcher Elemente könnte man ansonsten eine Oberfläche rendern die der Benutzer fühlen könnte. Jedoch ist diese Form der Ausgabe taktiler Informationen nicht besonders effektiv. Ein durchgeführtes Experiment[6] lieferte in dieser Richtung interessante Ergebnisse. So ist es sehr schwer statische Muster zu erkennen, welche auf diesen Schnittstellen ausgegeben werden. Die Methode, welche hier Verwendung findet, also mit Vibrationen zu arbeiten wird als viel effektiver bezeichnet.

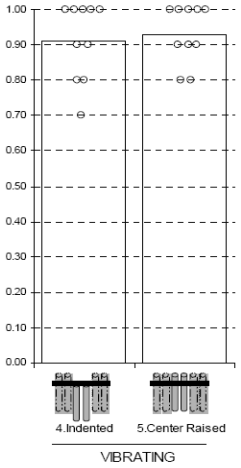
Jedoch geht die Studie [6] noch weiter und kombiniert statische Muster mit Vibrationen. Aufgrund der Ergebnisse kann man sehen, dass mit dieser Methode bestimmte Muster, hier nur simple „Einkerbungen“, sehr gut wahrgenommen werden können und eine Unterscheidung zu Ebenen ohne Einkerbungen möglich ist, man hier also Oberflächen vermitteln kann (Abbildung 4).

In diesem Aufbau wurden mehrere Stifte verwendet, die im einen Fall statisch eine Einkerbung in der Mitte bildeten, im anderen Fall vibrierten die Stifte um die versenkten in der Mitte. Wie zu sehen ist, wurde die Wahrnehmung im zweiten Fall deutlich besser.



[6]Abbildung 4: Simulierung von „Einkerbungen“

Würde man also laut dieser Studie statische Auslenkung durch Vibration vermitteln käme man auf exzellente Ergebnisse. Wie aber bereits gesagt, ist es mit der hier beschriebenen Technologie nicht möglich große Auslenkungen zu produzieren. Dazu gibt es aber ebenfalls interessante Erkenntnisse: Ein weiteres Experiment zeigte, dass es gar nicht notwendig ist, mit Höhenunterschieden zu arbeiten. Wirklich von Bedeutung ist nur die vibrierende Umgebung um eine Nichtvibrierende Stelle herum. Die Ergebnisse waren in beiden Fällen annähernd die gleichen: Die Benutzer spürten eine deutlich Einkerbung (Abbildung 5).



[6]Abbildung 5: Einfluss von Vibration und Höhendifferenzen

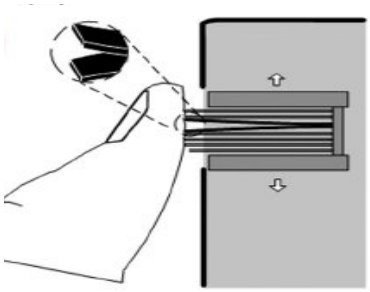
Natürlich nimmt eine Anordnung mit solchen Stäben ziemlich viel Raum ein und damit Vibrationen zu erzeugen ist aufwändig und benötigt viel Energie. Aus diesem Grund ist sie schlecht für mobile Anwendungen zu gebrauchen.

Ausgehend von diesen Ergebnissen wäre eine Kombination von mehreren der beschriebenen Aktoren deshalb ein durchaus interessantes Unterfangen, denn man könnte hiermit in recht kleiner Bauform solche Oberflächendarstellenden taktilen Interfaces realisieren, in dem man einfach viele dieser Elemente nebeneinander platziert und sie in unterschiedlichen Mustern in Schwingungen versetzt. In einer anderen Arbeit[9] wird gesagt, dass solche Aktoren sehr stark verkleinert werden könnten und so eine Schnittstelle bestehend aus vielen solcher Elemente in entsprechender Größe realisierbar scheint. Wenn das möglich sein sollte, ist es jedenfalls ein viel versprechender Weg.

3.2.Manipulation der Hautoberfläche

Abgesehen von der eben erwähnten Anwendung von sich biegenden piezoelektrischen Plättchen gibt es weitere Forschungsaktivitäten in dieser Richtung. Hier wird nicht versucht ein vibrierendes Interface durch diese Aktoren zu erzeugen, vielmehr wird die Biegung dazu benutzt, die Haut des Benutzers zu strecken, bzw. zusammenzudrücken.

Zu diesem Zweck werden eine Reihe dieser Aktoren nebeneinander Positioniert und über elektrische Impulse Biegen sich deren Enden entsprechend zueinander hin oder voneinander weg. Das Prinzip ist recht gut in Abbildung 6 zu erkennen.



[7]Abbildung 6: Funktionsweise der Schnittstelle

Durch die Kraft, die auf diese Art und Weise auf die Haut ausgeübt wird, entsteht der Eindruck unterschiedliche Formen und Strukturen zu berühren. Eine ähnliche Methode wäre es, in die Haut hineinzudrücken und somit einen Impuls auszulösen. Jedoch Ist dies, wie bereits zuvor erwähnt nicht einfach umzusetzen, währenddessen hier wieder die erprobten Aktoren zum Einsatz kommen können.[7]

Der Prototyp, welcher dieses spezielle Interface beinhaltet besteht aus einem Plastikgehäuse, welches auch ein LCD Display enthält. Die taktile Schnittstelle befindet sich auf der linken Seite an einer Position, an der der Daumen des Anwenders positioniert werden kann. Das Interface selber setzt sich in diesem Fall aus 8 piezoelektrischen Platten zusammen, die einzeln angesteuert die Kraft auf den Daumen des Benutzers übertragen können. Zu ersehen ist das gerät in Abbildung 7.

[7]Abbildung 7: Prototyp

Beugt man seinen Daumen nach oben oder nach unten kann man über die Schnittstelle Eingaben tätigen welche z.B. das Navigieren in einem Menü möglich machen. Durch Fingerdruck kann man seine Eingaben bestätigen.

Auch bei dieser Art der Stimulation bestätigt sich das zuvor Festgestellte: Sich bewegende Teile werden vom Anwender viel besser wahrgenommen als stillstehende. Aus diesem Grund bleiben die Lamellen nicht einfach stehen nachdem sie eine bestimmte Position erreicht haben, sondern sie bewegen sich auch hier kontinuierlich unter einem bestimmten Muster. Genau wie in den vorherigen Fällen ist es mit dieser Umsetzung ebenfalls möglich, Muster anhand von Vibrationen zu erzeugen. Die Lamellen können mit hoher Frequenz angesteuert werden und diese Frequenzen können zwischen den einzelnen Aktoren variieren. Mit dieser Methode können komplexere Muster erzeugt werden, als das es mit einem einzelnen Element möglich wäre. Dieses Verfahren ist ähnlich zu der in 3.1.4. vorgestellten Möglichkeit, jedoch sind die dadurch erzeugten Impulse nicht dieselben, da es sich hierbei nach wie vor eher um das Strecken und Dehnen der Hautpartien handelt als um das Vibrieren selbst. Jedenfalls ist es hier möglich eine Vielzahl unterschiedlicher „taktiler Icons“ zu rendern und darzustellen.

Ausgehend von dieser Anordnung sind aber noch weitere Stimulationsmuster möglich: Wenn sich alle Lamellen in Abhängigkeit der Zeit in richtiger Weise bewegen, kann man so die Illusion erzeugen, irgendetwas würde über den Daumen hinweg gleiten. Selbstverständlich kann man dieses Verhalten auch in Abhängigkeit zur Benutzereingabe setzen. Dieser tätigt also eine Eingabe, indem er seinen Daumen bewegt, und bekommt zur gleichen Zeit ein Feedback dazu. Ein Einsatzszenario das sich hier natürlich sofort ergibt wäre das scrollen in Dokumenten.

3.2.1.Bestimmung von Einflussfaktoren

Für den letzteren erwähnten Einsatzbereich gibt es hauptsächlich einen ausschlaggebenden Faktor, die maximale Geschwindigkeit, die solch eine „simulierte Bewegung von Strukturen“ haben konnte und gleichzeitig noch für den Anwender bemerkbar und klar Unterscheidbar war.

Zu diesem Zweck wurde ein Experiment durchgeführt, welches ein interessantes Ergebnis vorweisen konnte. Wurde die Bewegungsgeschwindigkeit erhöht, so nahm die Wahrscheinlichkeit ab, dass die Testpersonen die korrekte Bewegungsrichtung bzw. überhaupt eine Bewegung wahrnehmen konnten. Bei sehr hohen Werten erreichte man ein Erlebnis, welches mit der zuvor erläuterten unabhängigen Schwingungen der einzelnen Lamellen vergleichbar war, nichts mehr jedoch mit Bewegung in eine Richtung zu tun hatte. Die ermittelte höchstmögliche Geschwindigkeit bei der zufrieden stellende Resultate zu erwarten sind lag bei 0,34m/s.

Wie genau haben nun allerdings solche gerenderten Strukturen auszusehen? Auch hier wurde ein Experiment durchgeführt, bei dem eine Große Anzahl von Testpersonen einer Vielzahl unterschiedlicher taktiler Stimulationen ausgesetzt wurde Die variierten Parameter waren vor allem die Wellenformen der einzelnen Lamellen, die Intensität der Bewegung und die Geschwindigkeit, mit welcher sie sich bewegten. Die Ergebnisse belegen einen großen Einfluss der Wellenform auf die empfundenen Eindrücke. Die Einflüsse der Intensität und der Geschwindigkeit waren allerdings im Allgemeinen nicht klar einzuordnen. Erst nach Aufschlüsselung zu den einzelnen Wellenformen hin konnten klare Ergebnisse erzielt werden. So haben auch diese Parameter einen Einfluss auf das subjektive Empfinden, jedoch äußert sich das bei jeder Wellenform anders und mit einem nicht so hohen Einflussfaktor.[7]

Mit Hilfe dieser Erkenntnisse können nun unterschiedliche taktile Icons unter Rücksichtnahme auf die ausgelösten Empfindungen des Benutzers definiert werden. Es wurden hier nur recht einfache Muster verwendet um einen ersten Einblick in diese Thematik zu erhalten und einen guten Ausgangspunkt zu haben. Jedoch ist es unumgänglich hier noch weitere Forschung zu betreiben um genauer feststellen zu können, welche Parameter zu den besten Ergebnissen führen und schlussendlich zu einem umfassenden taktilen Vokabular zu gelangen.

3.2.2.Interaktionsmodalitäten

Durch Anwendung dieses Interfaces können nun unterschiedliche Arten von Interaktionsmodalitäten gefunden und definiert werden.[7]

Zum einen kann durch die Möglichkeit, diverse Icons zu rendern, eine Abgrenzung unterschiedlicher Ereignisse getroffen werden. Navigiert man durch ein Menü, so können die Menüpunkte jeweils durch ein anderes taktiles Icon dargestellt werden. Bekommt man einen Anruf, so können unterschiedlichen Kontakten ebenso unterschiedliche Icons zugeordnet werden. Hierzu ist allerdings zu sagen, dass dies auch mit den bereits beschriebenen Mechanismen möglich ist und die Umsetzung hier eventuell nicht unbedingt optimal ausgeführt ist. Um ein diesbezügliches Feedback zu erhalten muss der Anwender mit dem Interface in Kontakt stehen, was beim Navigieren durch ein Menü wahrscheinlich noch der Fall sein wird, beim Empfangen eines Anrufes jedoch hat man das Telefon selten in der Hand. Hier scheint mir die Verwendung von Vibrationen sinnvoller, da man das Gerät nur am Körper tragen muss und den Anrufer identifizieren kann auch ohne das Gerät heraus zu nehmen. Natürlich können die Muster bei dieser Methode viel akkurater abgebildet werden und eine genauere Einteilung unterschiedlicher Ereignisse ist jedenfalls gegeben. Die Frage ist nur inwiefern das im Falle von Anrufererkennung notwendig ist, schließlich muss sich der Benutzer auch zu jedem Kontakt die entsprechende taktile Signatur einprägen und das ist auch nur in einem begrenzten Ausmaß möglich.

Eine weitere Interaktionsmöglichkeit über solch eine Schnittstelle wäre das Scrollen in Dokumenten. Über das Interface können durch Daumenbewegungen Eingaben getätigt werden die dazu führen, dass innerhalb der entsprechenden Anwendung nach oben bzw. unten gescrollt wird. Während der Bewegung können taktile Informationen zum Benutzer zurück übermittelt werden in der Art und Weise, dass er die entsprechende Bewegung fühlt. Somit ist er immer über die aktuelle Bewegungsrichtung und Geschwindigkeit informiert, ohne sich von anderen wichtigen Tätigkeiten abwenden zu müssen. Dadurch, dass zusätzlich zur Bewegung auch noch unterschiedliche Icons dargestellt werden können, ist es auch möglich diverse charakteristische Stellen innerhalb von Dokumenten heraus zu heben. So kann sich die gefühlte Signatur beim Scrollen über einen Link oder ein Bild ändern und so das aktuell angezeigte Element signalisieren.

Dies ist wie gesagt ein sehr wichtiger und interessanter Ansatzpunkt für diese Technologie, jedoch gibt es auch hierbei Einschränkungen. So ist es nur möglich Bewegung in einer Dimension darzustellen. Für das Scrollen in Dokumenten mag das vielleicht ausreichend sein, jedoch stößt man damit an die Grenzen sobald es darum geht nicht nur nach oben und unten, sondern auch nach links und rechts zu wandern. Hier kommt noch der Faktor hinzu, dass die verwendeten Displays in mobilen Geräten recht klein sind und man rasch an einen Punkt kommt der einer seitlichen Verschiebung des Sichtfeldes bedarf.

Denkbar ist auch eine Anwendung im Bereich der Navigation. Man kann ein mobiles Gerät dazu benutzen, um zu einem bestimmten Punkt hin zu navigieren. Die heutigen Mobiltelefone sind häufig schon mit einem GPS System ausgestattet und mit Navigationssoftware ausgerüstet. Die Darstellung und Übermittlung der Richtungsinformationen kann nun ohne weiteres durch solch eine taktile Schnittstelle erfolgen. Mittels einer Bewegung am Daumen wird signalisiert, ob man in die richtige Richtung unterwegs ist, steht man falsch herum, so fühlt man eine Bewegung nach hinten. Dreht man sich, so wird diese Bewegung immer langsamer und verkehrt sich in eine Vorwärtsbewegung die immer schneller wird wenn man sich der korrekten Blickrichtung annähert. Doch auch hier erkennt man recht schnell die Grenzen dieser eindimensionalen Darstellung. Auch in diesem Fall würde sich eine Bewegung in mehrere Richtungen als äußerst nützlich erweisen.

Eine andere vorgestellte Interaktionsmöglichkeit liegt darin begründet, Statusinformationen über das Interface zu übermitteln. So kann man signalisieren ob ein Bekannter gerade online gekommen ist, sich der Batteriestatus geändert hat oder ähnliches. Das alles ohne seine aktuelle Tätigkeit, wie z.B. einen Telefonanruf zu unterbrechen. Aber auch hier ist der einschränkende Faktor der geforderte

direkte Kontakt zur Benutzerschnittstelle. Achtet man nicht darauf den Daumen richtig zu platzieren während man telefoniert, verpasst man die Information unweigerlich. Aus diesem Grund würde sich hier wie im ersten Fall eine Signalisierung über Vibration eher anbieten, da der Kontakt zum Gerät jedenfalls gegeben ist, der Kontakt zur kleinen Schnittstelle aber nicht unbedingt.

Bei genauerer Betrachtung wird also folgendes klar: Diese Art taktiler Interfaces ist gewiss sehr viel versprechend, doch in keiner der genannten Anwendungsgebiete wirklich ausgereift und verwendbar. Insbesondere die Beschränkung auf eine Dimension stellt ein großes Problem dar und es sollte versucht werden, das in nachfolgenden Forschungsprojekten zu beheben.

3.3.Elektrostatische Stimulation

Neben den eben beschriebenen Ansätzen, welche mittels Vibrationen versuchen taktile Eindrücke zu vermitteln, gibt es noch andere Forschungsaktivitäten, welche sich hierfür elektrischer Signale bedienen. Bei den viel versprechendsten Projekten wird allerdings nicht direkt mit Strömen gearbeitet. Man erzeugt vielmehr ein Elektrostatisches Feld, über welches Kräfte auf den Körper ausgeübt werden können. Laut der Aussage in [8] hat bereits Mallinckrodt in den fünfziger Jahren das folgende Phänomen beobachtet.

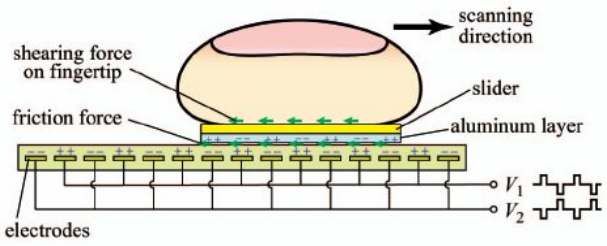
Setzt man eine von der Umgebung isolierte Metallplatte unter Spannung und berührt diese, so fühlt sich das gänzlich anders an, als bei einer spannungsfreien Platte. Bei berühren Der Platte bildet sich aus dieser und der leitfähigen Schicht im inneren des Fingers quasi ein Kondensator. Wird nun eine periodische Spannung angelegt, so ergeben sich elektrostatische Kräfte, welche den Finger an die Platte ziehen und auf diese Weiße Reibungskräfte erzeugen.

Dies ist eine Weitere Methode um in sehr kleinen Bauformen, wie sie im mobilen Bereich üblich sind, eine taktile Stimulation zu erzeugen. Schnittstellen dieser Art haben einen noch kleineren Platzbedarf als die piezoelektrischen Aktuatoren. Was man hier allerdings sehen muss, ist die Tatsache, dass man mittels dieser Methode keine Formen bzw. Stufen darstellen kann. Irgendeine Art von Erhebung darzustellen ist Aufgrund der technischen Funktionsweise nicht möglich. Das Einsatzgebiet hierbei beschränkt sich auf die taktile Darstellung von Oberflächenstrukturen[8].

3.3.1.Technische Umsetzungsschwierigkeiten

Neben technischen Umsetzungsproblemen kommt grundsätzlich noch erschwerend hinzu, dass die Wirksamkeit dieser Stimulation je nach den vorherrschenden Umgebungsbedingungen sehr stark variiert. Als besonders negativ hat sich hier Feuchtigkeit an der Fingeroberfläche erwiesen. Befindet sich nämlich ein Flüssigkeitsfilm zwischen Finger und Platte wirken die erzeugten Kräfte nur innerhalb des Films und nicht mehr, oder nur ungenügend auf den Finger selbst. Diese Eigenschaft erweist sich leider als Fatal, da man auf Grund dessen nicht in der Lage ist, immer die gewünschten Stimulationen so hervorzurufen wie man das beabsichtigt. Die Probleme treten besonders im Hinblick auf Anwendungen im mobilen Bereich zu Tage, da hier solche Einflüsse noch viel extremer Ausgeprägt sind.

Um trotz der genannten Widrigkeiten weiterzukommen, bediente man sich eines Hilfsmittels. Nicht der Finger selbst dient als Teil der elektrostatischen Krafterzeugung, sondern ein zweites, bewegliches Metallplättchen auf welchem man den Finger platziert. Die Kraft wirkt nun also auf einen zwischengelagerten Schieber welcher diese direkt an den Benutzer weiterleitet und somit unabhängig von Beschaffenheit und sämtlichen Eigenschaften der Hautoberfläche arbeiten kann. Abbildung 8 Illustriert diesen Aufbau [8].



[8]Abbildung 8: Prinzip der elektrostatischen Stimulation

Liegt an dieser Konstruktion keine Spannung an, so kann der Anwender den Schieber ohne zusätzlichen Kraftaufwand bewegen, sobald jedoch eine Spannung vorherrscht, wird es merkbar schwerer eine Bewegung in die entsprechende Richtung auszuführen. Arbeitet man zusätzlich noch mit sich periodisch umkehrenden Spannungen, so kehrt sich auch der Kraftaufwand im gleichen Maße um, was zu mechanischen Vibrationen und zum Eindruck, eine charakteristische Oberflächenstruktur zu überfahren, führt. In Abhängigkeit der Wellenform ist es möglich, verschiedenste Strukturbeschaffenheiten zu simulieren.

Parallel zur Realisierung solcher elektrostatischen Feedbacksysteme läuft natürlich ebenfalls die Entwicklung von Geräten, welche reale Strukturen abtasten und in entsprechende Signale umwandeln können um sie danach möglichst naturgetreu wieder auszugeben. Diese Forschungsaktivitäten fallen aber nicht in die konkrete Themensetzung dieser Arbeit und werden daher nicht näher beleuchtet.

3.3.2. Experiment

Um die Wirksamkeit solcher Interfacesysteme zu evaluieren wurden Experimente durchgeführt. Konkret ging es darum, mehrere reale Oberflächenstrukturen aufzuzeichnen und über die taktile Schnittstelle wieder aus zu geben. Die Testpersonen mussten dann die realen Muster den Simulierten zuordnen. Die Erkennungsrate bei dieser

Prozedur betrug knapp 80% Wie man in Abbildung 9 erkennen kann, handelte es sich hier um Unsicherheiten zwischen ähnlichen Strukturen. Es gab keine Verwechslung mit einer komplett anderen Oberfläche, was sicherlich auf ein großes Potential dieser Technologie schließen lässt.

Abweichungen traten hauptsächlich dadurch auf, dass die Geschwindigkeit und der ausgeübte Druck auf die Oberfläche sowohl bei den Testpersonen als auch beim Abtastgerät variierten und dadurch Ungenauigkeiten produziert wurden[8].

		Percentage of answers				
		a	*b*	*c*	*d*	*e*
Presented material	*a*	**95**	5	0	0	0
	b	15	**65**	0	5	15
	c	5	20	**75**	0	0
	d	0	0	5	**80**	15
	e	0	0	0	20	**80**

[8]Abbildung 9: Ergebnisse

3.3.3.Kombination mit LCD

Die hier vorgestellten Elemente können zum einen äußerst dünn ausgeführt und zum anderen, mit den entsprechenden Materialien, transparent hergestellt werden.

Dadurch ergibt sich eine Kombinationsmöglichkeit, wie sie auch schon in 3.1.3 vorgeschlagen wurde: die Integration in herkömmlichen LCD oder Touchscreen Displays. Man legt und befestigt also solch ein dünnes transparentes Interface über einem Monitor und bekommt anschließend sowohl visuelles als auch taktiles Feedback über das kombinierte Medium. Mittels dieser Umsetzung erhält man ein fortschrittlicheres und vielseitigeres Interface als durch die oben vorgeschlagene Lösung mit vibrierenden Displays, jedoch sieht man sich auch größeren, vor allem technischen, Umsetzungsschwierigkeiten entgegen. Durch die zwischen Touchscreen und Anwender befindliche Schichte ergeben sich Probleme bei der Erkennung der Aktuellen Mausposition und im Aktuellen Prototypen ist es auch nur möglich Eingaben zu tätigen, solange man den Finger möglichst parallel zur Display Oberfläche positioniert[8].

3.3.3.Interaktionsmodalitäten

Integriert man diese elektrostatischen Schnittstellen als eigenständige Teile in mobilen Geräten, so ergeben sich ähnliche Interaktionsmodalitäten wie in den bereits erläuterten Fällen. Da man hier, wie auch im Falle des Dehnen und Streckens der Haut unterschiedliche Oberflächenstrukturen Simulieren kann, kommen auch ähnliche Anwendungsfälle in Betracht. Der Punkt, an dem sich diese Technologien unterscheiden, ist sicher die Tatsache, dass hier der Benutzer seinen Finger über das Interface hinweg bewegen muss, um taktile Informationen zu erhalten. Im anderen Fall konnte man den Finger ruhen lassen, und dennoch mit dem Gerät interagieren.

Aus diesem Grund erscheinen für diese Art von Interaktion die übrigen Interfacesysteme als besser geeignet. Betrachtet man allerdings die Kombination mit Touchscreens so eröffnen sich vielfältige Möglichkeiten. Mit Hilfe dieser Technologie wäre es denkbar, verschiedenen Widgets am Schirm verschiedene Oberflächen zuzuordnen. Das wiederum hätte den Vorteil, dass sich der Anwender nicht ununterbrochen auf das mobile Gerät konzentrieren müsste wenn er eine Eingabe tätigt, sondern anhand der gefühlten Oberflächen am Schirm navigieren könnte. Das zuvor dargelegte Beispiel der Nummerntasten auf einem Touchscreen könnte so ebenfalls umgesetzt werden.

Hierbei sind allerdings die technischen Einschränkungen zu beachten. Abgesehen von der bisher unzuverlässigen Eingabeerkennung dieser Systeme gibt es zurzeit noch eine extreme Einschränkung: Wie oben erwähnt, ist der direkte Kontakt zur Haut äußerst unzuverlässig und durch viele Faktoren beeinflusst. Die Eingabe geschieht demnach über eine Platte zwischen Benutzer und Interface. Optimiert wurde das ganze dadurch, dass man diese durch einen Aufsatz ersetzte, ähnlich einem Fingerhut. Der Anwender wird jedoch nach wie vor dadurch belastet, eine zusätzliche Komponente verwenden zu müssen, möchte er taktiles Feedback erhalten. Dies ist ein nicht unerheblicher Einschnitt in den Benutzerkomfort und wird die Verbreitung solcher Systeme gewiss beeinträchtigen[8].

Vom momentanen Stand aus gesehen wäre also für recht einfache Implementationen, wie das genannte Touchscreen-Tastenfeld, die einfachere Lösung mittels vibrierender Displays vorzuziehen, da die vielseitigeren Einsatzmöglichkeiten hier nicht zur Wirkung kommen.

3.4.Taktile Touchscreens

Aufbauend auf der Arbeit aus 3.1 wurde aus den eben genannten Gründen versucht, die im Grunde einfache Idee der Kombination aus piezoelektrischen Aktuatoren und Touchscreen Displays weiter zu führen. Man bedient sich hierbei der Tatsache, dass diese bereits in äußerst kleiner Bauform hergestellt werden können. Ähnlich wie in der vorgestellten Methode der elektrostatischen Stimulation integriert man die Aktuatoren gewissermaßen innerhalb der Displays. Die Darstellung von visuellen Elementen wird dadurch nicht beeinträchtigt. Legt man eine Spannung an, dehnen sie sich aus und drücken die oberste transparente Schichte ein wenig nach oben. Die Auslenkungen betragen hier weniger als 0,05mm.[9]

Die entscheidende Verbesserung besteht darin, dass nicht länger die gesamte Displaykomponente bewegt werden muss, sondern nur ein sehr kleiner und leichter Teil davon. Dazu kommt die ausgesprochen gute Dämpfung dieser Anordnung. Findet eine Bewegung innerhalb des Displays statt, so ist das außen am mobilen Gerät nicht zu bemerken. Der Anwender spürt also tatsächlich nur die taktile Stimulation an seinem Finger und nicht einen Teil davon auch noch in seiner Hand.

Grundsätzlich wurde mit drei unterschiedlichen Wellenformen gearbeitet, mit denen die piezoelektrischen Elemente angesprochen wurden. Die ersten beiden simulierten beim Anwender das Gefühl eines Klickens. Im einen Fall empfindet man dieses als etwas härter, im anderen Fall als etwas weicher. Mittels dieser Formen kann dem Benutzer ein entsprechendes und vor allem natürliches Feedback beim Betätigen eines Buttons geliefert werden. Die dritte Wellenform ist eigentlich eine Kombination aus den ersten beiden und suggeriert ein elastisches Empfinden. Damit lassen sich sehr gut Übergänge simulieren.

Das hier vorgestellte Verfahren stellt eine interessante Weiterentwicklung der Vorgängerarbeiten dar und hat auf Grund der Einfachheit des Konzepts und der dennoch hohen Wirksamkeit sicherlich ein großes Potential. Vom technologischen Standpunkt aus scheint es auch unkritischer zu sein, als das elektrostatische Verfahren, welches hier noch mit enormen Schwierigkeiten zu kämpfen hat.

4. Abschließende Diskussion

Im Laufe dieser Arbeit wurden eine Reihe viel versprechender Möglichkeiten aufgezeigt, die Welt der mobilen Endgeräte durch die Einbeziehung der taktilen Dimension zu bereichern. Dies sind sicherlich Schritte in die Richtige Richtung, da vor allem im mobilen Bereich solch ein zusätzlicher Informationskanal als äußerst Wertvoll zu betrachten ist.

Die Methoden, mit denen versucht wurde, diese Aufgabenstellung zu bewältigen sind äußerst vielfältig und reichen von Vibrationen, über Dehnung und Zusammenziehung von Hautpartien bis hin zur Verwendung elektrischer Ströme. Die wichtigste Entdeckung in diesem Zusammenhang scheinen hierbei wohl die piezoelektrischen Aktuatoren zu sein, welche sich in immer anderer Art und weiße in diversen Entwicklungen eingesetzt finden. Durch diese Elemente eröffneten sich unzählige Möglichkeiten im mobilen Bereich, da sie sehr klein gebaut werden können und einen Spannungsbereich haben, welcher auch einsetzbar ist. Andere Technologien haben leider mit diversen Problemen zu kämpfen, wie die elektrostatische Stimulation. Ein grundsätzlich viel versprechender Weg, solange jedoch zusätzliche einschränkende Faktoren vorhanden sind, wie in diesem Fall der Fingeraufsatz, wird sich diese Variante trotz breitem Einsatzspektrum gewiss nicht durchsetzen.

In diesem Kontext interessant ist die Tatsache, dass einige Einflussfaktoren bisher nicht, oder nur ungenügend genutzt wurden. So gibt es nicht viel Forschungsaktivität bei der Verwendung von Strömen um dadurch Sinneseindrücke zu vermitteln. Abgetan wird das in einigen Arbeiten mit der Begründung dadurch recht leicht ein unangenehmes Empfinden auszulösen. Vielleicht könnten aber auch hier durch bessere Verfahren gute Ergebnisse erzielt werden. Ein anderer Faktor der nirgends erwähnt wird ist die Temperatur. Keines der gefundenen Systeme verwendete diese Art der Stimulation um Informationen an den Benutzer zu übermitteln, jedoch kann das Temperaturempfinden einen nicht zu unterschätzenden Einfluss auf die Empfindung von taktilen Reizen haben. [10]

Es werfen sich allerdings noch einige Fragen auf, welche nicht nur spezifische, sondern alle genannten taktilen Elemente betreffen. Sämtliche Arbeiten, welche solche Interfaces behandeln und in die mobile Sparte einzuordnen sind lassen ein paar Fragen außen vor, welche aber gerade hier behandelt werden sollten. Der Energieverbrauch solcher Schnittstellen wird nicht oder nur ungenügend beleuchtet. Hier handelt es sich aber um einen nicht zu unterschätzender Faktor bei Geräten, welche mit solchen Komponenten ausgestattet werden sollen. Inwieweit verringern sich Akkulaufzeiten in Geräten mit taktilen Schnittstellen und Ist der Anwender bereit, aufgrund des gewonnenen Komforts und der bereicherten Interaktionsmöglichkeiten, auf eine längere Einsatzdauer zu verzichten?

Weiters ist insbesondere im mobilen Bereich die Frage nach der Lebensdauer essentiell. Wie verhalten sich diese neuen Entwicklungen unter rauen Bedingungen, wie sie ohne Zweifel vorherrschen? Liegt die Betriebsdauer bis zu deren Versagen in dem gleichen Rahmen wie die, der übrigen Komponenten? Das sind Fragen die jedenfalls beantwortet werden müssen, sollten sich diese Interfaces durchsetzen wollen. Natürlich spielt hier auch die Kostenfrage mit ein. In solch einer frühen Phase wird es sich für den Endbenutzer nicht lohnen hier zu investieren, aber auf was für ein Niveau könnten sich die Kosten einpendeln, auch diese Frage wäre weitere Untersuchungen wert.

Ein in jeder Arbeit behandelter, und dennoch unbeantworteter Punkt ist weiters in der Erstellung geeigneter Stimulationsmuster zu finden. Alle Forschungsprojekte begnügen sich für ihre Arbeit mit einfachen Wellenformen um einfache taktile Eindrücke zu erzeugen. Sehr interessant scheint aber die Suche nach komplexeren Mustern und gewissermaßen dem Aufbau eines taktilen Vokabulars.

Nach diesen vielen, vor allem technischen, Fragen stellt sich noch eine Konzeptionelle. Welche Möglichkeiten gibt es, Content für solch eine taktile Welt zu generieren? Die Standardfunktionalitäten werden natürlich recht einfach abgedeckt, ein Button ist ein definiertes Objekt, dem einfach eine taktile Signatur entsprechend der verwendeten Technologie zugeordnet werden kann. Möchte man nun aber individuell seine Generierten Inhalte und/oder Programme mit taktiler Information anreichern so fehlen zurzeit entsprechende Werkzeuge, mit deren Hilfe man das umsetzen könnte. Hier ist jedenfalls noch viel Raum für weitere Forschungsansätze zu finden.

REFERENCES

1. Jonathan C. Roberts, Sabrina Paneels, "Where are we with Haptic Visualization?" whc, pp. 316-323, Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07), 2007.
2. Wikipedia. http://de.wikipedia.org/wiki/Taktil
3. Salisbury, K. Conti, F. Barbagli, F. “Haptic rendering: introductory concepts” pp. 24- 32 Computer Graphics and Applications, IEEE, 2004
4. Poupyrev, I., Maruyama, S. and Rekimoto, J. “Ambient Touch: Designing tactile interfaces for handheld devices.” pp. 51-60, Proc. IST02, ACM, 2002
5. A. Nashel and S. Razzaque, “Tactile virtual buttons for mobile devices,” pp. 854–855, CHI ’03 extended abstracts on Human factors in computing systems.ACM Press, 2003.
6. María Oyarzábal, Masashi Nakatani, Robert D. Howe, "Vibration Enhances Geometry Perception with Tactile Shape Displays," pp. 44-49, Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07),2007
7. Luk, J., Pasquero, J., Little, S., MacLean, K. E.,Levesque, V. and Hayward, V. “A Role for Haptics in Mobile Interaction: Initial Design Using a Handheld Tactile Display Prototype.” pp. 171-180 Proc. of the ACM 2006 Conference on Human Factors in Computing Systems, CHI 2006., 2006.
8. Yamamoto, A., Nagasawa, S., Yamamoto, H. and Higuchi,T. “Electrostatic Tactile Display with Thin Film Slider and Its Application to Tactile Tele-Presentation Systems.” pp. 209-216, HIGUCHI Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST), 2004
9. Pouyrev, I.,Maruyama, S., “Tactile interfaces for Small Touch Screens.” pp 217-220, Proc. UIST 2003, ACM Press (2003)
10. Vasilios G. Chouvardas, Amalia N. Miliou, Miltiadis K. Hatalis, “Tactile Displays: a short overview and recent developments” International Conference on Automation and Tectnology, ICTA'05, 2005

Visual Mobile Interaction

Daniel Felsing
dfelsing@edu.uni-klu.ac.at
0161052

ABSTRACT
Handheld Computer und speziell Mobiltelefone sind in der heutigen Zeit zu einer natürlichen Alltagserscheinung geworden. Der technologische Trend entwickelt sich in die Richtung immer mehr Ressourcen in einen immer kleiner werdenden Formfaktor zu integrieren. Die gebotenen Möglichkeiten eröffnen ein enormes Potential und lassen solche Geräte zu einem vielversprechenden Anwärter als Interaktionsmedium im bevorstehenden ubiquitären Zeitalter werden. Auch der Einsatz von größeren Displays, über die wir Informationen über unsere Umgebung erhalten ist heutzutage schon weit verbreitet. Meine Arbeit soll sich mit Systemen zur mobilen visuellen Interaktion auf größeren Displays beschäftigen. Es gibt hier eine Vielzahl an unterschiedlichen Lösungsansätzen, von denen einige in dieser Arbeit vorgestellt und diskutiert werden.

EINLEITUNG
Mark Weiser hatte die Vision des „Ubiquitous Computing". Der Begriff wurde das erste Mal im Jahre 1991 in seinem Artikel „The Computer of the 21st century" verwendet [1]. Seine These war, dass die profundesten Technologien diejenigen seien, die sich dem Sichtfeld des Anwenders entziehen. In seiner Vision verschwindet der Computer als Gerät also mehr und mehr. Stattdessen soll die Informationstechnik unser tägliches Umfeld durchdringen, mit dem Ziel Mensch und Computer nahtlos miteinander zu verbinden. Als Synonyme hören wir heut zu tage auch Begriffe wie „Pervasive Computing", „Calm Technology" oder auch „Everywhere Computing". Man erhofft sich mit dieser Idee eine möglichst unsichtbare Integration von Computern in alltäglich verwendete Gebrauchsgegenstände unserer Umgebung, um einen natürlichen Umgang mit informationsverarbeitenden Geräten zu realisieren, der stets zum jeweiligen Kontext passt. Weiser hat auch darauf hingewiesen, dass die wirkliche Kraft nicht in einem einzelnen dieser Medien verborgen liegt, sie entsteht erst durch die Interaktion all dieser.

Handheld Computer und speziell Mobiltelefone sind in der heutigen Zeit zu einer natürlichen Alltagserscheinung geworden. Der momentane technologische Trend entwickelt sich in die Richtung immer mehr Ressourcen in einen immer kleiner werdenden Formfaktor zu integrieren. Der Großteil der mobilen Geräte ist mittlerweile mit integrierten Kameras, Sensoren, hochauflösenden Farbdisplays und einer Vielzahl von Schnittstellen nach außen wie Bluetooth, WLAN oder auch Infrarot ausgestattet. Sie haben mittlerweile schon die Möglichkeit unsere Umgebung zu sehen, abzutasten und abzuhören [2]. Die Menschen tragen ihre Mobiltelefone und PDAs auch ständig mit sich. Dank dem Einzug von immer mehr Technik in diese kleinen Begleiter, finden sie zunehmend Verwendung in der Erfassung als auch der Speicherung von persönlichen und geschäftlichen Daten (z.B. für Photos, Notizen aber auch Office Materialen). Somit stehen wir gewissermaßen ständig im Kontext mit unserer Umgebung. Die gebotenen Schnittstellen können dazu verwendet werden, um sich mit anderen in unserer Umgebung befindlichen Geräten zu verbinden.

Auch der Einsatz von größeren Displays an öffentlichen Plätzen, über die wir Informationen über unsere Umgebung erhalten ist immer weiter verbreitet. Es wäre wünschenswert diese Technologien in Zukunft interaktiver und für mehrere Benutzer gleichzeitig zugänglich zu machen. Die Kommunikation ist nämlich meist leider in eine Richtung orientiert, da für ein interaktives System meist spezielle Hardware von Nöten wäre oder der Schutz vor Vandalismus eine höhere Priorität besitzt. Wenn man daran denkt wie sehr unsere Art Präsentationen abzuhalten durch die Einkehr digitaler Projektoren und Office Präsentationen beeinflusst wurde, ist auch hier ein enormes Potential erkennbar, mehr Interaktionsmöglichkeiten einzubinden. Digitale Whiteboards ermöglichen es beispielsweise mehreren Benutzern mit digitalen Informationen über Gesten und Berührungen zu interagieren, um so Kollaborationen zu ermöglichen. Diese kosten jedoch im Vergleich zum restlichen Equipment heutiger Büro- und Konferenzräume ein kleines Vermögen. Aber auch im privaten Bereich haben größere Bildschirme längst Einzug gefunden und so ebnet sich langsam der Weg für eine breite Masse an innovativen Anwendungen, vor allem wenn man auch den heutigen Grad der Vernetzung mit dem globalen Internet in Betracht zieht.

Diese mobilen Endgeräte erlangen dafür einen besonderen Stellenwert, da sie abgesehen von ihrer Allgegenwärtigkeit auch ein eigenes „privates" Informationsdisplay besitzen, welches sie für ein Szenario eines „gemeinsam genutzten" Bildschirms sehr attraktiv werden lässt. Zusätzlich beinhalten sie, wie bereits erwähnt, auch eine Vielzahl von Daten, die für die Kommunikation als auch Interaktion sehr relevant erscheinen. Die genannten Fakten eröffnen ein enormes Potential und lassen solche Geräte zu einem möglichen Anwärter als Standardinterface im bevorstehenden ubiquitären Zeitalter werden. Um ein Fundament an Interaktionsparadigmen hervorbringen zu können, ähnlich wie bei beim Desktop Computer Maus und

Keyboard das WIMP Paradigma (Windows, Icons, Menues, Pointers) erschaffen haben, müssen jedoch erst intuitive und effiziente Interaktionsmechanismen gefunden werden, die auf eine ubiquitäre Umgebung anwendbar sind [2].

Es gibt eine Reihe von interessanten Ansätzen, um mittels Mobiltelefonen oder PDAs mit größeren Displays interagieren zu können. Hier soll auch der Fokus gesetzt werden. Es sollen vor allem die Ideen jener Arbeiten näher beleuchtet werden, die ohne teure Zusatzhardware auskommen. Zuerst möchte ich jedoch noch ein paar wichtige Aspekte rund um diese Thematik aufzeigen.

ARTEN VON DISPLAYS

Die Interaktion mit größeren Displays kann grob in die Anwendungsbereiche Personal, Semi-Public und Public unterteilt werden [3]. Persönliche Displays sind nur für die Nutzung im privaten Rahmen gedacht und erlauben einem einzelnen Benutzer große Mengen an Informationen zu visualisieren und zu verarbeiten. Semi-Public Displays findet man meist in Bereichen mit Zutrittsbeschränkungen wie zum Beispiel Büro- oder Konferenzräumen Sie sollen es mehreren Menschen ermöglichen gemeinschaftlich auf einem Bildschirm zu arbeiten. In Abbildung 1 wird der iRoom der Stanford University gezeigt, in dem es möglich ist mit mehreren Benutzern über eine interaktive Oberfläche multimodal zu kollaborieren. Wie der Name schon sagt findet man Public Displays an öffentlichen Orten. Diese sind meist in Zonen situiert an denen auch sehr viele Passanten zu finden sind oder wenn mit höheren Wartezeiten zu rechnen ist, wie es beispielsweise bei U-Bahn Stationen, Flughäfen, Busstationen oder auch Einkaufsmärkten der Fall ist. Abbildung 1 zeigt ein solches Display in einer U-Bahn Station in Wien [4][5].

Abbildung 1. (Links) iRoom in Stanford, (Rechts) U-Bahn in Wien [4][5]

In [11] wird von einem enormen Zuwachs von Displays in unserer täglichen Umgebung gesprochen. Leider werden diese jedoch vielfach zu Werbezwecken, und damit um unser tägliches Konsumverhalten zu kontrollieren, verwendet. Man bedenke hier die riesigen Reklametafeln in heutigen Großstädten, die marktführende Konzerne an vielbesuchten Plätzen installieren (siehe Abbildung 2). Nachdem „öffentliche“ Plätze die Spiegelbilder unserer Kultur und unseres sozialen Umfelds sind, stellt sich die Frage, ob der momentane Weg der richtige ist. Dieser Platz könnte viel zu einem lebendigeren Kulturkreis als es heute der Fall ist beisteuern. Trotzdem ist eine enorme Balance zwischen Inhalt, Ort und Interaktionsmöglichkeiten nötig, um selbst nach Umstrukturierungen einen positiven Nutzen ziehen zu können, ohne störend oder unangenehm zu wirken wie es derzeit der Fall ist. Gerade in Anbetracht unserer technologischen Möglichkeiten ist es also Zeit für neue kreative Visionen, um sozial orientierte Inhalte hervorzubringen.

Abbildung 2. Unser Leben in der „Screenworld“ [11]

DESIGNKRITERIEN

In [3] werden eine Reihe von Design Kriterien beschrieben, die für Interaktionen mit größeren Displays von Bedeutung sind.

Serendipity

Der Begriff, der auf Deutsch so viel wie glücklicher Zufall bedeutet, beschreibt die Möglichkeit eines Benutzers spontan mit einem großen Display interagieren zu können. Ein hohes Maß an „Serendipity“ weist darauf hin, dass ein Benutzer mit einem willkürlichen Display sehr leicht spontan interagieren kann.

Portability

Sie soll aufzeigen wie wahrscheinlich es ist, dass jemand ein bestimmtes Instrument, welches für die Interaktion notwendig ist, mit sich trägt. Interaktionsmechanismen, die nur Dinge benötigen, die man immer am Körper trägt, inklusive dem Körper selbst, haben eine hohe Portabilität. Mobiltelefone und PDAs sind sehr portabel und die meisten von uns tragen sie stets mit sich.

Dexterity

Das Kriterium beschreibt wie einfach es einem Benutzer fällt, eine Interaktionstechnik anzuwenden. In einem Supermarkt wäre es beispielsweise umständlich beide Hände zu verwenden, wenn man eine davon benötigt, um eine Plastiktüte zu tragen.

Multi-User

„Multi-User“ Fähigkeit ist besonders in öffentlichen und halböffentlichen Plätzen wichtig. Dieses Kriterium ist ein Maß für die Möglichkeiten eines Systems, Funktionalitäten mehreren Benutzern gleichzeitig zur Verfügung zu stellen. Am Beispiel einer digitalen Anschlagtafel auf der Gegenstände zum Verkauf angeboten werden können, wäre es unter anderem wünschenswert, dass man sich ähnlich wie beim Gegenstück in der realen Welt, nicht anstellen muss, um ein Offert einzustellen.

Information Security / Privacy
Diese Aspekte sollen den Grad an Sicherheit bzw. Datenschutz wiederspiegeln. Sensitive Daten wie Namen oder Telefonnummern sollen nicht auf Grund irgendeiner Störung des Systems auf dem Display erscheinen.

Social Acceptability
Ob sich eine Interaktionstechnik eignet in Gegenwart anderer angewandt zu werden, ohne andere zu behindern oder sich dabei blamieren zu müssen, wird als „Social Acceptability" beschrieben.

Intentional vs. Unintentional Interaction
Im ersten Fall will der Benutzer aktiv eine Interaktion auslösen. Im Gegensatz dazu wäre es noch möglich, dass ein System die Anwesenheit eines Benutzers selbst erkennt und die Verbindung aufbaut.

INTERAKTIONSTECHNIKEN
Die heutzutage am weitesten verbreitete Interaktionstechnik ist direkt auf der größeren Oberfläche zu arbeiten. Eingesetzte Technologien reichen hier von kapazitiven Oberflächen, über DViT (Digital Vision Touch) Systemen, bis hin zu Ultraschall Techniken. Kapazitive Oberflächen registrieren Berührungen durch Spannungsänderungen durch die auf entsprechende Koordinaten auf dem Bildschirm geschlossen werden kann. Man interagiert hier entweder mit der eigenen Hand oder stiftähnlichen Gegenständen. Das kommerziell erhältliche SMART Board der Firma Smarttech [20] gibt es nun auch in einer Version die auf Digital Vision Touch setzt.

Abbildung 3. Mimio Virtual Ink Whiteboard [21]

Hingegen bekommt man beim Mimio Virtual Ink System [21] (siehe Abbildung 3) einen Satz an Stiften mit unterschiedlichen Farben, einen Radierer als auch eine spezielle Maus, die als Zeigegerät bei Präsentationen verwendet werden kann. Die Stifte senden bei Berührung mit dem Display ein Ultraschall Signal aus, durch dessen Messung die Position auf dem Whiteboard bestimmt werden kann.

Die Umsetzungen sind technisch sehr unterschiedlich. All diese Systeme verwenden jedoch als Interaktionsparadigma das direkte 1 zu 1 Mapping und somit sind sie sich hinsichtlich der Anwendbarkeit sehr ähnlich und weisen dieselben Vor- und Nachteile auf. Gerade wenn man an zukünftige Displaytechnologien wie OLED [8] denkt, mit denen man Größen von mehreren Metern ohne Probleme erreichen kann, stößt man beispielsweise im öffentlichen bzw. halböffentlichen Bereich mit den direkten Interaktionsmechanismen schnell an seine Grenzen. Ein großes Manko der vorher erwähnten Systeme ist nämlich die Skalierbarkeit. Ein Benutzer hätte ständig damit zu kämpfen nahe genug am Display zu stehen, um damit interagieren zu können, und gleichzeitig weit genug entfernt zu sein, um einen Überblick zu behalten. Touchscreens hätten zwar eine hohe „Serendipity", sind aber im öffentlichen Bereich wegen hygienischen Aspekten nur mit Bedenken einsetzbar. Würde man ein spezielles Eingabegerät fordern, würde hingegen wieder die Portabilität darunter leiden. Zudem sind die bisher genannten Systeme nur bedingt „Multi-User" tauglich. Gesten und Spracherkennung bieten zwar viele Vorteile, wenn man aber an den Public und Semi-Public Bereich denkt, sind hier meist mehrere Menschen gleichzeitig anwesend. Dank auftretender Hintergrundgeräusche und mehreren Personen, die sich im Sichtfeld einer Kamera bewegen, ist es für Programme relativ schwierig gewollte von ungewollten Interaktionen zu unterscheiden. In Hinsicht auf die definierten Designkriterien schneidet die Spracherkennung bezüglich „Privacy" und „Social Acceptability" in diesen Bereichen ebenfalls nicht gut ab [3]. Kombinierte Ansätze, die Sprache und Gesten gemeinsam verwenden, bieten für den privaten Gebrauch und Bereichen in denen nicht viele Hintergrundgeräusche auftreten, recht gute Möglichkeiten. Die Komplexität der Interaktionsmechanismen und damit die Lernkurve für den Benutzer sollte jedoch niedrig gehalten werden [10].

Die Steuerung mit einem speziellen Gerät, also einer Art Fernbedienung, bietet ein weiteres klares Paradigma zur Interaktion. Eine Möglichkeit wäre es ein solches Gerät öffentlich zugänglich zu machen. Gerade im Public bzw. Semi-Public Bereich könnten diese aber Vandalismus oder Diebstahl zum Opfer fallen und es stellt sich auch die Frage wie das Problem der „Multi-User" Fähigkeit gelöst werden soll. Nicht zu vergessen würde hier auch Hygieneaspekte wieder zur Diskussion stehen [3].

Visuelle Mobile Interaktion
Die Benutzer Geräte wie Mobiltelefone oder PDAs, die sie immer bei sich tragen, verwenden zu lassen, würde obige Kritikpunkte entschärfen. Wie bereits erwähnt besitzen diese kleinen Helfer neben ihrem ubiquitären Charakter mittlerweile eine Vielzahl an Schnittstellen und Sensoren, die man für die Interaktion verwenden könnte. Hier bieten sich Möglichkeiten für die öffentliche Interaktion, Mehrbenutzerkollaborationen als auch Einsatzgebiete für den privaten Gebrauch. Es sei erwähnt, dass Mobiltelefone hinsichtlich der Kriterien Portabilität und Einfachheit in der Benutzung („Dexterity") den PDAs voraus sind, da sie einerseits einen viel kleineren Formfaktor besitzen und eine einhändige Bedienung für die Eingabe von Texten und anderen Manipulationstechniken voraussetzen.

Im folgenden Teil meines Artikels werde ich einige interessante Ansätze beleuchten, die Interaktionstechniken mit mobilen Geräten zu erarbeiten versucht haben.

ANKERBASIERENDE VERFAHREN

Bei ankerbasierenden Verfahren ist es möglich direkt auf bestimmte Ankerpunkte zu zielen, die Auswahl bleibt aber auf genau diese beschränkt

Spotcode User Interface

Die Arbeit in [12, 13] stellt das Spotcode User Interface vor, welches es erlaubt ein mobiles Endgerät dazu zu verwenden, mit Computern in der näheren Umgebung zu interagieren. Als Kerntechnologien werden sogenannte Visual Tags (in diesem Fall Spotcodes [23]) und Bluetooth für die drahtlose Verbindung zu einem in der Nähe befindlichen Computersystem verwendet.

Funktionsweise

Die visuellen Tags funktionieren ähnlich wie ein zweidimensionales Barcode System, welches von einem kameragestützten Gerät, in diesem Fall ein Handheld, in Echtzeit gelesen werden kann. Diese Codes können entweder aktiver oder passiver Natur sein. Im aktiven Fall werden diese beispielsweise von einem Rechner generiert und auf einem größeren Display angezeigt. Ein passiver Tag könnte auf ein Poster oder Blatt Papier gedruckt werden. Sie können dazu benutzt werden Informationen zu kodieren – in diesem Fall sind das zwei Blocks: der „Service-Identifier“ und ein „Data-Block“. Ersterer wird dazu verwendet den richtigen Bluetooth Dienst anzusprechen, der zweite kann für Informationen verwendet werden, die für die Applikation wichtig sind. Am Mobiltelefon ist nur die Installation einer allgemeinen Software zur Erkennung notwendig, welche die kodierten Informationen des Tags und Messdaten, wie die Koordinaten des Tags am Telefondisplay, die relative Rotation des Handhelds zum Code und ein Maß für die Distanz zum Code an einen verbundenen Bluetooth Dienst schickt. Eine schematische Darstellung ist in Abbildung 4 ersichtlich.

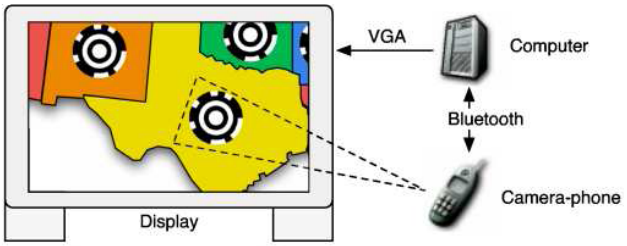


Abbildung 4. Funktionsweise des ankerbasierenden Ansatzes von [12]

Nachdem ein Benutzer auf einen Tag gezeigt hat, wird im Hintergrund automatisch eine Verbindung zu einem Server hergestellt und sämtliche ermittelten Daten inklusive Informationen über etwaig gedrückte Tasten übermittelt. Dies hält eine dahinterliegende Applikation ständig am Laufenden darüber, mit welchem Element ein Benutzer gerade interagiert. Jeder Code dient auf dem Display wie der Name schon vermuten lässt als Anker der Interaktion - im Gegensatz zu einem Cursor, mit dem man prinzipiell jeden beliebigen Pixel anvisieren könnte.

Elemente und Interaktionsmechanismen

In [12,13] werden insgesamt drei Interaktionsmechanismen vorgestellt. „Point“ bedeutet, dass man auf einen bestimmten Anker am Display zielt. Nach Auswahl durch einen Button am mobilen Gerät gibt es drei definierte Möglichkeiten darauf zu reagieren. Im einfachsten Fall würde irgendeine Aktion durch die Auswahl ausgelöst werden. Mit „Grasp“, wählt man einen bestimmen Tag für die Interaktion aus. Sinn und Zweck ist mit komplexeren UI Elementen, von denen einige in Abbildung 5 gezeigt werden, zu interagieren. Links ist ein Rotationselement ersichtlich, bei welchem man durch physisches Drehen einen Wertebereich einstellen kann. Einen Slider bedient man durch links-rechts Bewegung. Die dritte und letzte Möglichkeit benötigt das private Display des mobilen Gerätes. Durch „Throw“ wird ein willkürliches Benutzerinterface auf dem Handheld angezeigt. Die Anwendungsgebiete können hier vielfältig sein. Man könnte ein Listenmenü anzeigen, um beispielsweise eine Auswahl treffen zu können, oder auf eine Internetseite verweisen, auf der man eine Zahlung tätigt.

Abbildung 5. Spotcodes ermöglichen neuartige UI Elemente [22]

Anwendungsgebiete

Um die vorgestellten Mechanismen zu veranschaulichen wurden zwei Demoapplikationen entwickelt. Bei der World Map wurde ein einfaches Point & Click Interface implementiert, über welches ein Benutzer die Möglichkeit erhält Fluginformationen über eine Weltkarte (siehe Abbildung 4) einzusehen und Flüge zu buchen. Diese wird auf einem größeren Informationsdisplay angezeigt, wobei Spotcodes Regionen mit Flughäfen repräsentieren. Durch Auswahl zoomt die Anwendung in das gewählte Gebiet und offenbart einen genaueren Detailierungsgrad. So kann sich der Benutzer mit ein paar Klicks in ein Gebiet seiner Wahl hanteln. Schlussendlich bekommt er per „Throw“ eine Liste der Flughäfen auf sein Gerät geworfen und kann zuallererst seinen Startflughafen bestimmen. Das Spiel wiederholt sich bis ein entsprechender Zielflughafen selektiert wurde und der Benutzer kann bequem über sein privates Display relevante Fluginformationen einsehen und gegebenenfalls eine Buchung durchführen. Dies ist sogar ein sehr realistisches Szenario, da viele Mobilfunkbetreiber bereits die Möglichkeit bieten über das Telefon zu bezahlen. Ein

weiteres Anwendungsgebiet wäre beispielsweise das Schaufenster eines Reisebüros oder das bereitstellen eines virtuellen Schalters auf einem Flughafen.

Die komplexeren GUI Elemente wurden durch eine Audio-Jukebox Applikation veranschaulicht (siehe Abbildung 5). Die Auswahl der Musikstücke, Lautstärkenregelung und Vor-/Zurückspulen werden über die bereits vorher beschriebenen Spotcode Widgets realisiert.

Diskussion

Die Möglichkeit nur über bestimmte Ankerpunkte zu interagieren scheint mir persönlich intuitiv und einfach („Dexterity“) zu sein, solange die Applikation nicht zu viele davon verwendet und die Bedeutung dem Benutzer klar und deutlich vermittelt wird. Die Komplexität solcher Interfaces hält sich also sicherlich in Grenzen, wenn man ein gutes Maß an Benutzerfreundlichkeit erreichen möchte. Bei manchen Elementen momentaner Desktop Widgets optional Tag-Ankerpunkte für die Navigation mit mobilen Geräten anzubieten, wäre eine gute Idee. Bei Buttons, Slidern oder Menüs könnte ich mir das durchaus gut vorstellen. Es wäre sicher ein Gebiet bei dem es sich lohnen würde, weiter nachzuforschen. Auf „Multi-User“ Fähigkeit wird in der Arbeit nicht näher eingegangen. Durch die Entkopplung bestimmter Informationen auf das Display des Handhelds wird zumindest ein hohes Maß an Informationssicherheit und Privatsphäre geboten. Die Verbindung muss vom User aktiv aufgebaut werden.

Man bedenke bei dieser Arbeit, dass all dies nur auf einem aktiven Display möglich ist. Bei einem passiven, wie es z.B. bei einem Blatt Papier oder Poster der Fall wäre, funktioniert nur noch der „Throw“ Mechanismus. In der Beispielapplikation in Abbildung 5 bekommt man eine Liste der Titel des Spiderman Soundtracks auf sein Handy und kann dadurch eine mit einem Computer gekoppelte Stereoanlage bedienen [22].

Abbildung 5. Spotcode auf einem Blatt Papier [22]

Ein Vorteil dieser Codesysteme ist, dass die Erkennung anscheinend sehr robust ist (siehe das zerknüllte Stück Papier in Abb. 5). In der Arbeit wurde darauf hingewiesen, dass sie auch auf verschmutzten oder zerkratzten Displays noch gut funktionieren, wie es vor allem im Public bzw. Semi-Public Bereich der Fall sein kann. Ich persönlich befürchte jedoch, dass bei schlechten Lichtverhältnissen und einer geringeren Eigenhelligkeit des verwendeten Displays sehr wohl Schwierigkeiten bei der Erkennung auftreten könnten.

Visual Code Widgets

Einen anderen Weg solche Ankerpunkte für die Interaktion einzusetzen geht der Autor in [15].

Funktionsweise

Es wird eine verbesserte Version des Sony Cybercodes [16] eingesetzt, der sein eigenes lokales Koordinatensystem definiert, wodurch ein eindeutiges Mapping zwischen Code-Ebene (wirkliche Position) und Image-Ebene (auf dem Handheld Display) möglich wird. Zusätzlich können wie schon zuvor bei den Spotcodes beschrieben Messdaten wie die relative Rotation zum Gerät, die Distanz zum Code und die Koordinaten am Display bestimmt werden.

Elemente und Interaktionsmechanismen

Die Arbeit des Autors konzentriert sich zum größten Teil auf die Erarbeitung verschiedenster GUI Elemente mit dem Namen Visual Code Widgets (siehe Abbildung 6), mit dem Ziel eine Basis für zukünftige kreative Anwendungen zu schaffen. Abbildung 6 zeigt einen kleinen Ausschnitt aus dem zur Verfügung stehenden Spektrum. Es finden sich darin unter anderen verschiedenste Menüs für die Auswahl als auch Widgets für die Selektion (z.B. Check Boxen) und zur Dateneingabe (z.B. Sliders oder Dials). Sliders werden durch rechts-links Bewegung und Dials durch Rotation bedient. Sie eigenen sich für die Realisierung komplexerer mobiler Schnittstellen, und können schon wie bei der vorigen Arbeit entweder direkt auf Papier gedruckt oder auf einem größeren Bildschirm angezeigt werden. Der große Unterschied liegt jedoch darin, dass die Manipulation bzw. Interaktion auf dem privaten Bildschirm des mobilen Geräts passiert. Wie in Abbildung 6 ersichtlich wird die Auswahl mit dem durch die Kamera erfassten Bild überlagert, um den Zustand des Interaktionselements wieder zu spiegeln.

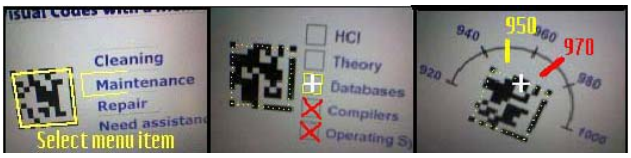


Abbildung 6. Visual Code Widgets [15]

In der Arbeit wird von zwei Modi der Interaktion gesprochen. Zum einen kann die Interaktion in „Echtzeit“ passieren. Dabei wird dem Schützen ein recht gutes Zielvermögen abverlangt, da das Zeigen auf ein Element und die Auswahl mit dem Steuerkreuz des Endgeräts parallel passiert. Im mittleren Bild der Abbildung 6 erkennt man das Problem recht gut. Man muss einerseits stets darauf acht geben, den Code nicht aus dem erfassten Bild der Kamera zu verlieren und gleichzeitig mit dem Fadenkreuz auf die Checkboxen zielen, was mir persönlich recht kompliziert erscheint. In der Arbeit wird jedoch darauf hingewiesen, dass noch zusätzliche Algorithmen zur Bildanalyse eingebaut werden sollen, die das Problem den Code ständig im Blickfeld haben zu müssen beheben soll – inwiefern hier die limitierten Rechenkapazitäten einen Engpass bilden bleibt aber fragwürdig. Sliders und Dials

können nur in diesem Modus verwendet werden – die Eingabe durch bspw. Drehen scheint mir in Anbetracht der genannten Probleme einfacher zu sein. Der zweite Modus wird als „Freezing“ bezeichnet. Der Benutzer muss ein Bild von dem Menüelement schießen und kann dann die Auswahl treffen. Dies behebt zwar das Problem, dass man nicht gleichzeitig Zielen muss – zum einen ist es so jedoch nicht möglich alle Widgets zu verwenden (siehe Sliders & Dials) und zum anderen muss es gelingen ein „wackelfreies“ Bild zu schießen, was mit vielen der heutigen integrierten Kameras gar nicht so leicht ist und sehr von den Lichtverhältnissen abhängig ist.

Um spontane Interaktionen zu realisieren, sind sämtliche Informationen wie der Typ des Widgets, das Layout und die Eigenschaften wie z.B. die Anzahl der Einträge eines Menüs im visuellen Tag kodiert. Zusätzlich enthält jeder Code noch Platz für allgemeine Informationen, um eine URL, Telefonnummer oder sonstiges abzuspeichern. Bei der vorherigen Arbeit wurden diese Daten an einen Dienst übermittelt, um Aktionen auf dem größeren Display auszulösen. Hier liegt der Fokus eher auf Anwendungen die am Handheld selbst laufen sollen, die Benutzerschnittstelle aber über das größere Display geliefert wird. Der Ansatz benötigt im Vergleich zum Spotcode Interface dafür speziellere Anwendungen am Handheld. Die Autoren haben in ihrer Arbeit eher den Fokus auf einen generischen Softwarekern zur Erkennung und Generierung der Widgets gesetzt.

Anwendungsgebiete

Als Einsatzgebiet würden sich beispielsweise Quizze anbieten, die meist in den Werbepausen nur kurz eingeblendet werden. Der Benutzer müsste sich nicht mehr die Telefonnummer als auch Fragen notieren und noch dazu eine entsprechende SMS verfassen. Stattdessen könnte er eine Anwendung starten, den „Freeze“ Modus verwenden, um ein Bild des gezeigten Menüs zu schießen und anschließend in Ruhe eine Auswahl treffen. Für einen automatischen Versand über die SMS Schnittstelle könnte die Telefonnummer im visuellen Tag codiert worden sein[15]. Aber auch bei größeren Informationsdisplays im Public bzw. Semi-Public Bereich könnte ich mir einen Einsatz vorstellen.

Diskussion

Der Erfolg von Anwendungen hängt meiner Meinung nach stark davon ab wie gut die semantische Bedeutung der GUI über diese Widgets vermittelt wird. Da nur das private Display des Handhelds verwendet wird, ergibt sich solange keine aktive Manipulation am größeren Display passiert eine gute „Multi-User“ Fähigkeit und „Privacy“. Die „Dexterity“ und die „Portability“ würde ich aber im Vergleich zu den Spotcode Interfaces insgesamt als schlechter einstufen, da einerseits der Echtzeitmodus viel zu kompliziert zu bedienen ist und andererseits für eine wirkliche Portabilität aus jetziger Sicht zu spezielle Anwendungen am Handheld benötigt werden.

VISUELLE BEWEGUNGSABTASTUNG

Hier wird im Gegensatz zu den ankerbasierenden Ansätzen meist versucht eine Maus nachzuahmen, um die Lernkurve dank des bewährten Prinzips besonders niedrig zu halten. Man interagiert also nur über Positionierung eines Cursors am Bildschirm. Die Berechnung erfolgt über die Erkennung von bewegten „features“ – also Anhaltspunkten, durch die ein Mapping passieren kann. Einen interessanten Weg gehen die Autoren mit dem in [17] vorgestellten C-Blink System.

C-Blink

Funktionsweise

Das System verwendet blinkende Lichtsignale, die über das LCD Display des mobilen Gerätes ausgesendet und von einer stationären Kamera aufgezeichnet werden. Einerseits dient dies dazu, dem größeren Display Informationen über getätigte Aktionen (z.B. Click) bzw. die Identität des mobilen Terminals zu übertragen. Andererseits lassen sich so auch Bewegungsveränderungen des Handhelds vor der fixen Kamera bestimmen, die dazu genutzt wird den Mauszeiger an eine korrespondierende Position am großen Display zu setzen (Abbildung 7). Andere konventionelle Übertragungsverfahren verwenden oft sehr schnelle LEDs und spezielle Sensoren für die Übertragung. Die Bildschirme mobiler Terminals kommen nicht einmal annähernd an die Blinkrate solcher speziellen Bauteile heran. Das Signal sollte aber trotzdem sehr robust sein, da bei der Verwendung von Kameras eine Vielzahl von optischen Störungen auftreten können.

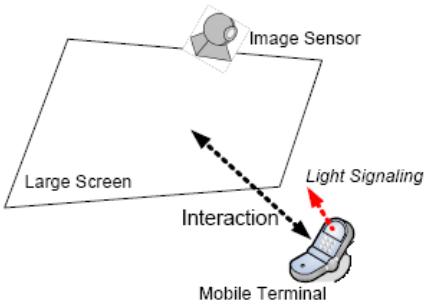


Abbildung 7. Funktionsweise von C-Blink (Erkennung über spezielle Lichtsignale durch eine stationäre Kamera) [17]

Bei diesem Ansatz wird die Information in sogenannte „hue-differences“, also in die Unterschiede mehrerer aufeinanderfolgender Farbtöne, kodiert. Mit entsprechend großen Toleranzen soll so ein recht robustes visuelles Signal erzeugt werden, welches auch bei Störungen, wie einem stark bewegten Hintergrund, und schlechteren Lichtverhältnissen noch detektiert werden kann. Wichtig ist dabei, dass die Kamera das Signal in der gleichen Frequenz abtastet, wie das LCD Display die Lichtsignale emittiert. Für den optionalen Austausch von Daten, der bei zwei im nächsten Absatz beschriebenen Interaktionstechniken verwendet wird, ist eine drahtlose Verbindung über

Bluetooth vorgesehen, um mit dem Management Server, auf dem auch die Applikation des größeren Bildschirms läuft zu kommunizieren. Der Benutzer muss nur eine allgemeine Software auf dem Handheld installiert haben, die in der Lage ist die benötigten Blinksignale auszusenden.

Elemente und Interaktionsmechanismen

Man interagiert bei diesem System nun wie folgt. Das Gerät fängt zu Beginn an ein spezielles Handshake Blink-Muster auszusenden, um die Verbindung zu initialisieren – theoretisch könnte hier auch eine Authentifizierung (z.B. über ein Passwort, das visuell zur stationären Kamera übertragen wird) stattfinden. Dazu muss, wie in Abbildung 7 ersichtlich, das Gerät mit dem Bildschirm in die Richtung des Sensors gedreht werden. Falls vom Handheld unterstützt wird jetzt auch die drahtlose Verbindung im Hintergrund initialisiert. Ein Cursor zur Navigation erscheint an einer Position, die davon abhängt wo im Blickfeld der Kamera das Lichtsignal detektiert wurde. Dies wird in Abbildung 8 recht gut veranschaulicht.

Abbildung 8. Links - Testaufbau, Rechts - Erkennung des C-Blink Signals [17]

Insgesamt werden drei verschiedene Interaktionstechniken, die auch in einer Beispielanwendung umgesetzt wurden, vorgestellt. Dabei handelt es sich um „Click“, „Grab“ und „Pitch“. Ausgelöst werden diese durch Drücken einer entsprechenden Taste des Telefon Keyboards, wodurch sich das Blinkmuster ändert und die Aktion auf dem größeren Display durchgeführt wird. „Click“ wird hauptsächlich für die Navigation verwendet, um zum Beispiel einen Button anzuwählen. „Grab“ bedeutet, dass man gerne ein Objekt der Begierde über den Wireless Kanal auf seinen Handheld laden möchte. Durch „Pitch“ kann man ein zuvor gewähltes an die Position des Cursors auf den Schirm transferieren.

Kombiniert man diese beiden Interaktionsmöglichkeiten erhält man eine ähnliche Funktionalität wie bei „Pick & Drop“, welches in [14] näher vorgestellt wird. Es handelt sich dabei um eine besondere Form des bekannten Drag & Drop Verfahrens und ermöglicht die Verschiebung von Datenobjekten zwischen zwei berührungsempfindlichen Displays (also z.B. zwischen zwei PDAs). Hier werden also Daten zwischen zwei Geräten direkt verschoben. „Grabbing & Pitching“ verwendet speziell Mobiltelefone, um Datenobjekte über den Umweg größerer Displays zu verschieben. In der von den Autoren in [17] gezeigten Beispielapplikation wurde eine Bildaustauschplattform entwickelt, mit der es möglich ist Bilder von einem Handheld aus an eine gewünschte Stelle zu platzieren und umgekehrt wieder herunter zu laden.

Probleme & Lösungsversuche

Probleme gibt es momentan noch beim Datendurchsatz des Blinksignals, was eine Cursorverzögerung > 500ms zur Folge hat. Eine Erhöhung der Abtastfrequenz würde aber zusätzliche Erkennungsschwierigkeiten bringen. In der momentanen Version funktioniert diese innerhalb eines Abstands von 30cm bis 1,2m vor der stationären Kamera bei mäßigen Lichtverhältnissen bzw. visuellen Störungen in Innenräumen noch recht problemlos. Im Freien sind die Ergebnisse jedoch gravierend schlechter ausgefallen. Nachdem die Cursorposition in Relation zum detektierten Blinksignal angezeigt wird, suchen die Autoren auch nach Möglichkeiten dem Benutzer zu vermitteln, wo sich das Sichtfeld der Kamera, also der Bewegungsspielraum für die Interaktion, liegt. Als Lösung wird vorgeschlagen auf dem Boden vor dem größeren Display eine Box aufzuzeichnen, in die sich eine Person während der Interaktion stellt, oder umgekehrt das Sichtfeld über automatisches Zooming & Panning der Position des Users anzupassen. Als mögliches Feedback dafür, welches aus der Sicht der Autoren auch für die Multiuser Interaktion interessant erscheint, wäre anstatt eines Cursors einen Videospiegel zu integrieren, ähnlich wie es in vielen Arbeiten zum Thema Virtual Reality diskutiert wurde (siehe Abbildung 9). Diese Überlegungen befinden sich aber noch in einem äußerst experimentellen Stadium. Die zukünftige Arbeit liegt unter anderem darin, an den aufgezeigten Problemen zu arbeiten und weitere interessante Techniken für die Interaktion hervor zu bringen [17].

Abbildung 9. Visuelles Feedback durch einen Videospiegel [17]

Anwendungsgebiete

Ein mögliches Anwendungsgebiet für ein Gerät mit integriertem GPS wäre das größere Display, welches unter anderem irgendwo öffentlich zugänglich gemacht werden könnte, als Navigationssystem zu verwenden. Man könnte sich eine Route planen und anschließend die Karte per „Grab“ auf seinen Handheld laden, auf welchem ein entsprechendes Programm angestoßen wird.

Denkbar wäre „Grabbing & Pitching“ auch in Kombination mit den ankerbasierten Ansätzen umzusetzen – wenn man an die World Map denkt [12] (Planung von Flugrouten). Spotcodes würden sich auch für die Fotoaustauschplattform eignen. Man könnte hier den Platz auf dem größeren Display z.B. in mehrere Slots unterteilen, die jeweils mit

einem Anker ausgestattet sind. Hier hat man auch das Problem nicht ständig im Sichtfeld einer stationären Kamera sein zu müssen. Ein Vorteil dieses Ansatzes wäre jedoch, dass die Hardwareanforderungen so minimal sind, dass nicht einmal eine integrierte Kamera benötigt wird.

Diskussion

In Hinsicht auf das Design Kriterium „Dexterity" würde ich C-Blink als eher mittelmäßig beurteilen. Wenn man Abbildung 9 ansieht, oder sogar noch von einer größeren Fläche ausgeht, sieht es nicht so einfach aus den ganzen Bildschirm mit einer einhändigen Nutzung des Geräts zu erreichen, außer man bewegt sich während der Interaktion vor dem Schirm hin und her. Den Handheld zwischen beiden Händen hin und her zu jonglieren, um das Sichtfeld der Kamera abdecken zu können, scheint auch nicht besonders befriedigend. Das Zeigeprinzip wirkt für einfache Anwendungen auf öffentlichen Terminals meiner Meinung nach einfach geeigneter. Wenn man diesen Umstand auch bei der „Multi-User" Fähigkeit in Betracht zieht, so würden sich Personen hier wohl oder übel in die Quere kommen. Der Ansatz mit dem Spiegel würde zwar das Feedback Problem bei einem bzw. mehreren Benutzern lösen, aber nicht das des beschränkten Platzes vor dem Schirm. Für ein kreatives Mehrbenutzerspiel könnte ich mir diesen vielleicht noch vorstellen, aber nicht für eine wirklich nützliche Anwendung. Um das System wenigstens zwei oder drei Benutzern gleichzeitig zugänglich zu machen, könnte man vielleicht mehrere eingezeichnete Rechtecke am Boden platzieren und auch dementsprechend das Sichtfeld der Kamera „virtuell" aufteilen. Hier wäre aber sicher eine Kamera mit höherer Auflösung notwendig, da sonst die Cursorbewegungen auf dem großen Display wohl noch ungenauer werden würden. „Privacy" wird hier nur schwer umzusetzen sein, da das private Display schon für das Blink Signal verwendet wird und somit nicht mehr so einfach für die Anzeige von Daten, die nicht für die Öffentlichkeit bestimmt sind, verwendbar ist. Auch werden in der Arbeit nur wenige Techniken zur Interaktion mit GUI Elementen beschrieben.

Direct Pointer

Funktionsweise

Die Verwendung des Handhelds wurde in dieser Arbeit einem Laserpointer nachempfunden. Ziel ist wieder den Cursor auf dem größeren Bildschirm zu manipulieren. Ähnliche Arbeiten beschäftigen unter anderem damit echte Laserpointer für die Interaktion mit größeren Displays zu verwenden. Diese bedienen sich aber meist speziellem Equipment und können nicht einfach mit mobilen Geräten verwendet werden [18, 19]. Im Gegensatz zum C-Blink System wird hier die integrierte Kamera des Handhelds verwendet (siehe Abbildung 10). Hier funktioniert die Bewegungserkennung ähnlich wie ein geschlossener Regelkreis – also über ständiges Feedback durch die eingefangenen Bilder. Die Handheld Kamera zeichnet also ein Bild auf und lokalisiert darin die Position des Cursors. Das Feedback wird dann an den Server geschickt, der abhängig vom Offset zum Mittelpunkt des Kamera Frames eine korrigierte Cursor Position am größeren Display errechnet. Den genauen Algorithmus dafür findet man in [13].

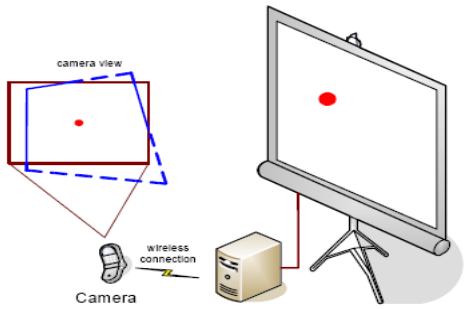


Abbildung 10. Direct Pointer - Laserpointer für aktive Displays [13]

Performanz

Die Autoren wollten in dieser Arbeit belegen, dass Ansätze über visuelle Bewegungsabtastung genauso effektiv sein können, wie solche mit speziellen Hardwarelösungen. Aus diesem Grund wurde in einer anschließenden Studie untersucht wie effektiv dieser Ansatz ist, indem die der Prototyp nach ISO9241-9 evaluiert und mit anderen Eingabegeräten aus der Literatur verglichen wurde.

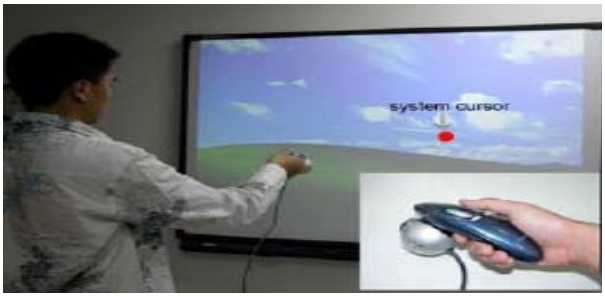


Abbildung 11. Prototyp [13]

Die ersten Versuche wurden aus diesem Grund noch nicht auf einem Handheld mit drahtloser Verbindung realisiert, sondern mit einer festverdrahteten Logitech Quickcam, die in der Lage ist mit 30 Bildern pro Sekunde aufzuzeichnen. Für die Click Operationen wurde eine Drahtlose Maus an der Kamera angebracht (Abbildung 11). Die Testpersonen standen bei den Versuchen 3,6 Meter von einer 2 Meter breiten Leinwand entfernt, auf die mit einer Auflösung von 1024x768 Pixel projiziert wurde. Dabei ergaben sich ähnliche gute Ergebnisse wie bei technisch intensiveren Verfahren [13].

Die zukünftige Arbeit liegt darin den Algorithmus weiter zu optimieren und die Funktionalität auf Handheld Geräte zu migrieren. Mit einem Nokia N93, welches ebenfalls in der Lage ist mit 30fps aufzuzeichnen würden sich wohl ähnlich gute Ergebnisse erzielen lassen – nur ist leider nicht jedes Handy so leistungsfähig - auf durchschnittlichen Geräten

werden wohl wieder Verzögerungen auftreten. Wie gut der Ansatz bei unterschiedlich auftretenden Lichtverhältnissen funktioniert, wurde in der Arbeit leider nicht erwähnt.

Anwendungsgebiete
Ein Einsatzgebiet für Präsentationen als Ersatz für einen Laserpointer im Semi-Public Bereich könnte vielleicht ein Anwendungsgebiet sein, da hier sowieso meist die Räume abgedunkelt werden. Inwiefern der Ansatz für den Public Bereich taugt müsste evaluiert werden und kann vom jetzigen Standpunkt aus nicht beantwortet werden.

Diskussion
Das Design Kriterium „Dexterity" scheint auf Grund des einhändigen Zeigeprinzips sehr gut erfüllt zu werden. Da mehrere Benutzer parallel einen solchen Zeiger auf dem Display bedienen können, würde einer Verwendung bei Mehrbenutzerkollaborationen im Semi-Public Bereich nichts im Weg stehen. Aber auch bei Präsentationen könnten mehrere Personen ihr Handy als Zeigegerät verwenden. Sollte jemand eine Frage zu einem Themengebiet haben, könnte derjenige einfach sein persönliches Gerät zücken und auf entsprechende Punkte zeigen („Multi-User").

HYBRIDE ANSÄTZE

Hybride Ansätze beschränken sich nicht auf Ankerpunkte oder Bewegungsabtastung und lassen sich demzufolge nicht so einfach klassifizieren. Es wird dabei versucht den visuellen Inputkanal vielfältiger zu nutzen, um komplexere Manipulationen zu ermöglichen. Zwei Arbeiten, die mir interessant erschienen sind, möchte ich nun in diesem Kapitel diskutieren.

Sweep und Point & Shoot

Die Techniken beschränken sich in diesem Beispiel zwar auf Mobiltelefone – würden mit ein paar Modifikationen aber auch auf anderen Handheld Geräten funktionieren. Voraussetzung ist hier wieder nur eine integrierte Kamera und eine Schnittstelle wie Bluetooth oder WLAN. Das Linke Bild in Abbildung 13 zeigt das Mapping auf den Joystick des Telefons.

Abbildung 13. Links – Mapping der Funktionalitäten, Rechts - Sweep [7]

Funktionsweise und Interaktion bei Point & Shoot
Bei dieser Technik, die in Abb. 12 ersichtlich ist, zielt der Benutzer auf ein entsprechendes Objekt auf dem größeren Bildschirm – in diesem Fall ein Puzzleteil. Das Display des Telefons dient dabei als Sucher, der kontinuierlich aktualisiert wird und ein Fadenkreuz zum Zielen besitzt. Der Punkt der Aufmerksamkeit des Users liegt nun rein auf dem kleinen privaten Display. Der Cursor auf dem größeren Display ist in diesem Augenblick nicht in Funktion. Die Interaktionstechnik wird durch Drücken der horizontalen Joystick Buttons ausgelöst (links für normales Klicken, rechts für Rotation und Drag & Drop). In dem Moment, in dem „Shoot" ausgelöst wird, erscheint kurz ein Raster von visuellen Tags auf dem Bildschirm, wie sie schon bei den Visual Code Widgets vorgestellt wurden. Auf Grund des eindeutigen Mappings zwischen Code- und Imageebene wird es so möglich präzise festzustellen wohin man auf den Schirm gezielt hat. Anschließend wird die Selektion des Objekts durchgeführt.

Abbildung 12. Point & Shoot [7]

Funktionsweise und Interaktion bei Sweep
„Sweep" hingegen setzt auf Abtastung des „Motion Flows", also des Bewegungsflusses. Die Auswertung der sequentiell aufgenommenen Bilder passiert aus Skalierbarkeitsgründen („Multi-User") im Gegensatz zum Direct Pointer auf dem Gerät selbst. Dabei werden die Differenzen zwischen diesen zur Errechnung einer relativen Bewegung verwendet, die dann zum größeren Display übertragen werden. In diesem Fall ist es egal wohin das Auge der Kamera zeigt, da keine Informationen vom größeren Bildschirm gebraucht werden. Um die Interaktionstechnik auszulösen hält man eine vertikale Taste des Joysticks gedrückt (nach oben für die normale Bewegung, nach unten für Rotation bzw. Drag & Drop), und kann dann ähnlich wie mit einem Besen in die gewünschte Richtung „fegen". Während man „fegt" kann man auch optional die Taste wieder loslassen, um eine bequemere Position einzunehmen und den Vorgang durch erneutes Drücken fortsetzen. Ausgewertet wird die X- und Y-Richtung als auch optional eine Rotation um die Z-Achse (siehe Abbildung 13, rechts).

Kombination von Sweep und Point & Shoot
Zusätzlich ist es möglich diese Techniken zu kombinieren – das Mapping ist wie bereits erwähnt Links in Abbildung 13 ersichtlich. Ist bei „Point & Shoot" kein Objekt im Fadenkreuz des Suchers wird durch Drücken nach links der Cursor absolut positioniert. Befindet sich ein Objekt im Sucher entspricht dies einer Auswahl. Durch Drücken nach rechts kann man wie bei Drag & Drop ein gewähltes Objekt an eine neue absolute Position schieben – je nachdem (z.B. bei einem Puzzlespiel) könnte man durch Drehung der Kamera auch gleichzeitig eine Rotation durchführen. Ein Drücken nach oben löst eine reine Mauskontrolle via „Sweep" aus und man kann den Cursor verschieben.

Drücken nach unten erlaubt das momentan selektierte Objekt durch „fegen“ zu verschieben [7]. Man könnte also beispielsweise ein Objekt „absolut“ auswählen und dann per „Sweep“ verschieben oder rotieren.

Insgesamt versucht man also mehr Informationen aus dem optischen Kanal herauszuholen, als es z.B. bei C-Blink, Direct Pointer der Fall ist und die Interaktion nicht nur auf Ankerpunkte wie beispielsweise bei den Spotcode Interfaces zu beschränken.

Performanz

Die Autoren haben in [7] auch eine detaillierte Analyse der Interaktionstechniken vorgestellt. Die „Sweep“ Technik hat hier schlechter abgeschnitten als „Point & Shoot“ bzw. die Nutzung des Telefonjoysticks zur Mausbewegung. Ein Grund dafür ist, dass durch die Abtastung von „Motion-Flows“ nur sehr schwer auf eine Beschleunigung des Cursors geschlossen werden kann. Diesen Umstand versucht man zukünftig durch einen optionalen Einsatz von Accelerometern zu verbessern. Die Testpersonen haben sich auch sehr schwer getan, die richtige Technik über die Telefontastatur anzuwählen. Hier stellt sich also die Frage, wie man Funktionalitäten mit ähnlicher Semantik besser abbilden könnte. Noch ein Nachteil bei „Sweep“ war die hohe Reaktionslatenz des Zeigers von ca. 200ms. Grund dafür sind die begrenzten Rechenkapazitäten – bei C-Blink war man hier jedoch in Bereichen > 500ms. Auch bei guter Aktualisierungsrate den Cursor indirekt durch „Sweeping“ zu positionieren scheint mir aber bei zeitkritischen Anwendungen trotzdem nicht so geeignet – hier haben Techniken, die den Cursor direkt manipulieren eindeutig Vorteile (z.B. Direct Pointer).

„Point & Shoot“ ist hingegen sehr anfällig auf die Distanz zum Display. Einerseits sind bei größerer Entfernung vom Schirm Objekte kleiner und schwieriger zu selektieren, andererseits treten dann auch vermehrt Probleme bei der Erkennung auf. Diesen Umstand könnte man vielleicht durch Zoomen und Bildstabilisierung verbessern [7].

Diskussion

Das Design Kriterium „Serendipity“ wird recht gut unterstützt, da dank der visuellen Codes automatisch die Verbindung hergestellt werden kann. Die „Dexterity“ würde ich als mittelmäßig bewerten. Positiv ist, dass man eine komplett einhändige Nutzung voraussetzt, andererseits scheint die Belegung des Joysticks nicht von Vorteil zu sein. Auch die Finger dürften unter einer längerfristigen Benutzung leiden. Mehrbenutzer-Fähigkeit könnte man theoretisch durch unterschiedliche Mauszeiger realisieren – das Gitter, das bei Point & Shoot eingeblendet wird, stört aber mit Sicherheit. Die Autoren erwähnen aber in [7], dass in Zukunft Display Technologien erscheinen werden, denen es möglich ist Infrarot Signale auszusenden, die nur für die Kamera, aber nicht für den Menschen sichtbar sind. Auf das private Display wird in dieser Arbeit nicht weiter eingegangen, könnte aber theoretisch zur Anzeige bzw. Manipulation von Informationen genutzt werden.

Motion-Flow, Marker-Object, Marker-Cursor

In [6] werden drei Interaktionstechniken vorgestellt, die einige Ähnlichkeiten zum vorherigen Ansatz haben. Die Autoren versuchen aber jeweilige Stärken und Schwächen der Ansätze bestmöglich einzusetzen.

Funktionsweise und Interaktion bei Motion-Flow

„Motion-Flow“ korrespondiert mit der zuvor vorgestellten „Sweep“ [7] Technik. Beide Ansätze bieten dieselben Möglichkeiten, abgesehen davon, dass in dieser Arbeit die Interaktion nicht durch „fegen“ sondern einfaches Bewegen in die entsprechende Richtung vollführt wird. Beide haben dasselbe Problem, dass eben immer nur indirekt über einen Mauscursor interagiert wird.

Funktionsweise und Interaktion bei Marker-Object

Der „Marker-Object“ Approach versucht sich so wie „Point & Shoot“ dem Problem zu entledigen, ständig mit einem Cursor navigieren zu müssen, um mit einem Objekt in Interaktion zu treten. Anstatt des Netzes mit dem es theoretisch möglich ist jeden beliebigen Pixel anzuvisieren, verwenden die Autoren dieser Arbeit einen visuellen Tag pro Objekt. Die Marker sind zuerst unsichtbar, bis der entsprechende Modus angestoßen wird. Es werden alle visuellen Codes kurz angezeigt, was zur Bestimmung dient, welches Objekt ausgewählt wurde. Sichtbar bleibt wie in Abbildung 14 ersichtlich nur noch der der ausgewählten Karte. Im Unterschied zu „Point & Shoot“ bei dem Drag & Drop eher als „Cut & Paste“ Operation umgesetzt wurde, erlaubt dieser Ansatz wirkliches Drag & Drop mit gleichzeitiger Rotation. Einmal aufgehoben kann man also ein Objekt willkürlich über die Bildschirmebene ziehen. Ein Vorteil von „Point & Shoot“ bleibt natürlich immer noch, dass man prinzipiell jeden Pixel selektieren könnte – verschieben gelingt hier jedoch nur in Kombination mit der „Sweep“ Technik.

Abbildung 14 – Marker Object [6]

Funktionsweise und Interaktion bei Marker-Cursor

Schreiben und Zeichnen soll durch den „Marker-Cursor“ Ansatz effizienter gestaltet werden. Dadurch wird eine recht genaue Bewegungs- und Positionserkennung möglich, die speziell für diese Operationen dienlich ist. Durch die 6-DOF Informationen, die sich aus visuellen Tags extrahieren lassen (z.B. durch Abstandsmessung), wird hier auch ein Einsatz für 3D Manipulationen möglich und interessant [6].

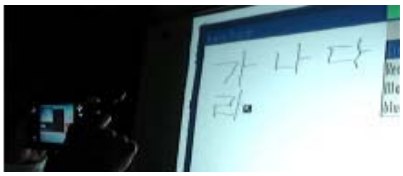


Abbildung 15 – Marker Cursor [6]

Performanz

Wie auch bei den vorherigen Forschungsprojekten ist aber auch dieser weit davon entfernt Bewegungen flüssig zu erkennen. Man hat die Performance durch Einsatz auf verschiedener Hardware gemessen. Einerseits ein Telefon, welches in der Lage ist mit ~15 Bildern pro Sekunde aufzuzeichnen und einen mobilen PC, der im Vergleich knapp 30fps schaffte. Auf dem schnelleren Gerät haben sich schon deutliche Fortschritte gezeigt – so lässt sich hoffen dass diese Methoden auch bald effektiv auf einem mobilen Endgerät genutzt werden können. Dies ist eine Frage der Zeit wie es scheint. Diese These bestätigt sich mitunter auch durch die Ergebnisse der Forschungsgruppe, die den Direct Pointer [13] erarbeitet hat, die nur mit einem Prototypen und noch nicht einem wirklichen Endgerät mit beschränkten Ressourcen gearbeitet haben.

Anwendungsgebiete

Die gezeigten Interaktionstechniken wurden in zwei Demoapplikationen veranschaulicht. Einer Schreibsoftware und einer Applikation zum Ansehen von Straßenkarten. Momentan werden Studien durchgeführt, wie effektiv sich diese Interaktionstechniken in anderen Applikationen wie z.B. zur 3D Manipulation einsetzen lassen. Ein potentielles Einsatzgebiet wäre auch die Mehrbenutzerkollaboration im Semi-Public Bereich. So hätten mehrere Benutzer die Möglichkeit auf dem größeren Schirm gleichzeitig zu schreiben, zu zeichnen oder Objekte zu transformieren bzw. zu verschieben. Gleichzeitig könnte man auch das private Informationsdisplay des Handhelds verwenden, um durch [17] „Pitching" Informationen auf das Display zu transferieren oder mit „Grabbing" wieder herunterzuladen. Eine zusätzliche Möglichkeit wäre auch per „Throw" [12] ein Benutzerinterface für weitere Optionen auf den Handheld zu werfen oder dieses für sonstige private Informationen zu nutzen. Es scheinen hier der Kreativität wirklich keine Grenzen gesetzt zu werden.

Diskussion

Das Design Kriterium „Dexterity" ist hier meiner Meinung nach besser, als beispielsweise bei Point & Shoot, da auf die komplizierte Tastenbelegung des Joysticks verzichtet wurde. Eine gute „Multi-User" Fähigkeit ist auf jeden Fall gegeben. Über die Nutzung des privaten Displays („Privacy") und die Einfachheit des Verbindungsaufbaus („Serendipity") wird in der Arbeit nicht näher eingegangen.

DISKUSSION UND AUSBLICK

Die diskutierten Ansätze bieten für mobile Endgeräte völlig neue Möglichkeiten. Sie könnten uns bei der Interaktion mit größeren Displays im privaten, public als auch semi-public in vielerlei Hinsicht neue Dimension offenbaren, ohne dafür spezielle Gegenstände für die Interaktion zu benötigen. Sehr viele Forschungsgruppen interessieren sich momentan dafür mobile Endgeräte als universelle Fernsteuerung einzusetzen. Nicht ohne Grund, denn wenn jeder wie in [3] beschrieben sein persönliches Gerät, welches in ständigem Kontext mit der Umgebung ist, mitbringen könnte und damit gleichzeitig vielschichtige Interaktionen möglich werden würden, so könnten bei entsprechender Kreativität der Anwendungen, eine Welle an neuen Möglichkeiten auf uns zu rauschen (siehe Abbildung16).

Abbildung 16 – Vision bald Wirklichkeit? [7]

Im privaten Bereich könnte man so vielleicht bald schon mit dem Mobiltelefon seinen DVD Player oder seine Stereo Anlage steuern. In Teil-öffentlichen Bereichen könnten bald Mehrbenutzerkollaborationen ohne teure digitale Whiteboards auskommen. Präsentationen, die mehr in eine Richtung orientiert waren, würden so interaktiver gestaltet werden, indem Personen einfach ihr Mobiltelefon zücken und sich aktiv beteiligen könnten (Fragen könnten so viel ausdrucksvoller an den Mann gebracht werden). Aber auch die interaktivere Gestaltung des öffentlichen Bereichs birgt viele Möglichkeiten.

Visuelle Tags werden momentan im asiatischen Raum immer weiter verbreitet, um die Realität virtuell zu erweitern und Interaktionen über Handheld Geräte zu ermöglichen. Erste Handyhersteller bemühen sich auch schon Software für die Erkennung schon bei Auslieferung bereit zu stellen. Die vorgestellten ankerbasierenden Ansätze zeigen vielversprechende Möglichkeiten, um mit öffentlichen Terminals oder privaten Displays interagieren zu können. Ein großes Problem ist meiner Meinung nach, dass es kein einheitliches Framework dafür gibt und eben eine spezielle Software am Telefon verwendet wird, auf die man sich erste einigen müsste, um wirkliche Portabilität zu ermöglichen.

Bei den Arbeiten, die auf visuelle Abtastung setzen scheint die Hardware jedoch noch nicht reif zu sein, da alle mit Verzögerungen des Mauszeigers und Problemen bei der Erkennung von Gesten auf Grund der beschränkten Ressourcen zu kämpfen haben. Die Verzögerung lag im besten Fall bei ca. 200ms, was noch immer viel zu hoch ist. Wie in [9] beschrieben hat System Lag nämlich einen multiplizierenden Effekt auf Fitt`s „Index of difficulty", welcher verwendet wird, um die menschliche Performance

bei Zeige- und Selektionsoperationen zu messen. Viele vorgestellte Interaktionstechniken zeigen zwar schon gute Ansätze – diese Stecken jedoch meist leider noch in den Kinderschuhen und sind von einem richtigen Einsatz noch recht weit entfernt, da die Praktikabilität in entsprechenden Anwendungen erst erforscht werden müsste. Es gibt auch sehr wenige Studien, die dies evaluieren. Sofern vorhanden habe ich versucht mögliche Ergebnisse, die von den Autoren angeführt wurden, in die Arbeit zu integrieren – leider waren diese aber nur sehr spärlich vorhanden.

Ich habe auch lange nicht alle Arbeiten behandelt, da dies das Ausmaß bei weitem gesprengt hätte – aber ich hoffe einen guten Einblick gegeben zu haben. Bei so vielen Ansätzen liegt es wohl daran erst einmal alle gewonnen Erkenntnisse auf einen Nenner zu bringen, um daraus erweiterte Konzepte für reale Anwendungen zu erschaffen. Es bleibt auch darauf zu hoffen, dass die Hardware bald soweit ist. So könnten diese Technologien bald in unserem Alltag Einzug finden. Der Trend zeigt aber, dass dies nicht in all zu ferner Zukunft der Fall sein könnte.

REFERENZEN

1. Mark Weiser. The computer of the 21st century. S*cientific American*, 265:94-104, September 1991.
2. Ballaghas, R., Borchers, J., Rohs, M., Sheridan, J.: The Smart Phone: A Ubiquitous Input Device. *Pervasive Computing-5-(1)*-(2006).70-77
3. Ballagas R., Rohs M., Sheridan J.: BYOD – Bring your own device.
4. Mark Patrick Baudisch, Nathaniel Good, and Paul Stewart: Focus plus context screens: combining display technology with visualization techniques. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 31–40. ACM Press, 2001.
5. E. A. Bier and S. Freeman. MMM: A user interface architecture for shared editors on a single screen. In *Proc. of the 4th Annual Symposium on User Interface Software and Technology (UIST'91*), pages 79–86. ACM, 1991.
6. Jeon et al.: Interaction Techniques in Large Display Environments using Hand-held Devices. In Proceedings of VRST 2006,November 2006, Limassol, Cyprus.
7. Rafael Ballagas et al. Sweep and point & shoot: Phonecam-based interactions for large public displays. In *Extended Abstracts CHI. ACM Press*, April 2005.
8. J. Shaw and P. Seidler. Organic electronics: Introduction. *IBM Journal of Research and Development*, 2001
9. MacKenzie, I. S., and Ware, C. Lag as a determinant of human performance in interactive systems. Proceedings of the SIGCHI conference on Human factors in computing systems.ACM Press (1993), 488–493.
10. Francis Quek, David McNeill, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, Karl E. McCullough, and Rashid Ansari. Multimodal human discourse: gesture and speech. *ACM Trans. Computer-Human Interaction.,* 9(3):171–193, 2002.
11. Struppek Mirjiam: The Urbane Potential of Public Screens for Interactions. In: *intelligent agent Vol. 6 No. 2, Special Issue: Papers presented at the ISEA2006 Symposium*, August 2006
12. Anil Madhavapeddy, David Scott, Richard Sharp, and Eben Upton. Using camera-phones to enhance human-computer interaction. In *Adjunct Proc. of Ubicomp. Springer-Verlag*, September 2004.
13. Hao Jiang et al: Direct Pointer - Direct Manipulation for Large-Display Interaction using Handheld Cameras, *Tsinghua University, Beijing, China and Microsoft Research Asia, Beijing*, 2006
14. Rekimoto, J. Pick-and-Drop: A direct manipulation technique for multiple computer environments. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST`97)*, 1997, pp. 31-39.
15. M. Rohs. Visual code widgets for marker-based interaction. In *Proc. IWSAWC'05: ICDCS 2005 Workshops,* June 2005.
16. J. Rekimoto and Y. Ayatsuka. CyberCode: Designing augmented reality environments with visual tags. In *Proceedings of DARE, Designing Augmented Reality Environments*.Springer-Verlag, 2000.
17. Kento Miyaoku et. al. C-blink: a hue-difference-based light signal marker for large screen interaction via any mobile terminal. In *Proc. UIST*, pages 147–156. ACM, 2004.
18. Olsen, D. and Nielsen, T. Laser Pointer Interaction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI2001)*, 2001, pp.17-22.
19. Vogt, F., Wong, J., Fels, S. and Cavens, D. Tracking multiple laser pointers for large screen interaction. *Extended Abstracts of the ACM Symposium on User Interface Software and Technology* (UIST2003), 2003, pp. 95-96.

INTERNET

20. SMART Board – http://www.smarttech.com
21. Mimio Virtual Ink – http://www.mimio.com
22. Anil Madhavapeddy et al.: Spotcode Research http://www.cl.cam.ac.uk/research/srg/netos/uid/spotcode.html
23. Spotcodes – http://www.spotcode.com

Adressen zuletzt am 13. Juni 2007 besucht.

Navigation in großen Dokumenten auf kleinen Bildschirmen

Marcel Bricman, Bakk. Techn.
Universität Klagenfurt,
kci:design & development
Kohldorferstraße 98, 1-23
mbricman@kci-group.com

ABSTRACT
Die Verwendung von großen Dokumenten auf kleinen Bildschirmen erzeugt einen Bedarf an alternativen Navigationsmöglichkeiten, der von heute gängigen Implementierungen nur mangelhaft erfüllt wird. Diese Arbeit gibt einen Überblick über Technologien, die das Ziel haben, solche Möglichkeiten zu erschließen bzw. auch jene die das Potential dazu hätten. Insbesondere wird auf die Anwendungs- Situation auf mobilen Endgeräten eingegangen, die einerseits weitere Restriktionen mit sich bringt, aber andererseits auch gewisse Techniken erst möglich macht, wenn jene beispielsweise die Beweglichkeit des Geräts voraussetzen. Die einzelnen Techniken werden in einer Bewertungsmatrix nach Kriterien und Anforderungen verglichen, die sich aus den Mängeln von klassischen Implementierungen ergeben.

Author Keywords
Dokument- Navigation, Benutzerschnittstellen für mobile Geräte, kleine Bildschirme

ACM Classification Keywords
H.5.2 User Interfaces (D.2.2, H.1.2, I.3.6): Graphical user interfaces (GUI).

EINLEITUNG
Große Dokumente werden üblicherweise in scrollenden Schnittstellen-Elementen dargestellt. Diese zeigen einen Ausschnitt des Dokuments und erlauben es dem Benutzer mit diversen Methoden, das Dokument unter diesem Ausschnitt zu verschieben. Dieser Ansatz der Navigation in großen Dokumenten hat zwar gewisse Ähnlichkeit zur realen Welt in der wir auch nur einen Ausschnitt der Realität wahrnehmen können. Die übliche Umsetzung vermisst allerdings feine, kontinuierliche Wahrnehmung der Position und deren Änderung, was den Benutzer beim schnellen Durchblättern bald orientierungslos mach oder frustriert. Eine visuelle Suche im Dokument wird durch die stark wechselnde Darstellung behindert. Der Fokus dieser Arbeit liegt auf mobilen Geräten mit kleinen Bildschirmen, bei denen der beschriebene Effekt natürlich noch gravierender ist. Es werden Lösungsansätze präsentiert, die es Benutzern erlauben, effizient in großen Dokumenten zu navigieren, aber es wird auch auf Risiken und Probleme der jeweiligen Lösungen eingegangen.

Techniken des Umgangs mit scrollenden Elementen umfassen Scrollbalken, Tastaturbefehle, Verschieben (Dragging) mit 2D- Eingabegeräten, Hardware- Scrollräder und andere.

Als Lösungsansatz wird vor allem Speed Dependent Automatic Zooming (kurz SDAZ) [IH00] sowie Verbesserungen [CSW05] davon besprochen, das die Vorteile von Zooming Interfaces ohne deren Nachteil der komplexen Interaktion mit Scrollen kombiniert indem der Geschwindigkeitsparameter eines beliebigen Interaktionsmodus variabler Geschwindigkeit als Parameter für die Größe der Darstellung verwendet wird.

Weiters werden auch folgende Methoden des Scrollen, sowie Kombinationen von diesen, vorgestellt: Scrollen mit eingebautem Bewegungs- oder Neigungssensor [EM04], Scrollen durch Verschieben des Ausschnitts anstatt dem Verschieben des Dokuments [LKM07], Scrollen mit eingebauter Kamera, Scrollen mit Übersicht / Navigationsansicht und weitere.

Den Kern der Arbeit bildet ein Vergleich der verschiedenen Methoden, der nach objektiven wie subjektiven Kriterien die Vor- und Nachteile der einzelnen Lösungen gegenüberstellt.

Abschließend finden sich Anregungen für die Generalisierung einiger Techniken und es wird eine kurze Einschätzung der weiteren Entwicklung der Verbreitung der vorgestellten Techniken gegeben.

DOKUMENTNAVIGATION

Die Darstellung von Inhalten die größer sind als der zur Verfügung stehende Bildschirmbereich wird üblicherweise durch das Anzeigen eines Ausschnitts des Dokuments realisiert, dieser Ansatz liegt allen vorgestellten Lösungen zugrunde, die Lösungen unterscheiden sich also hauptsächlich durch die Wahl des Ausschnitts und die angebotenen Interaktionsmodi.

Wir unterscheiden hier zwei Arten der Navigation in Dokumenten, die sich von der jeweiligen Intention des Benutzers herleiten. Diese verschiedenen Anforderungen werden durch verschiedene Interaktionselemente unterschiedlich gut unterstützt.

Konsumieren / Lesen

Der Benutzer konsumiert das Dokument in der aufgezwungenen linearen Weise, muss also den Ausschnitt der Darstellung nur so weit verschieben, dass er einen weiteren Teil sinnvoller Größe des Dokuments erkennen kann. Üblicherweise sprechen wir hier von *Lesen*, wollen aber durch diese Vereinfachung nicht auf das Medium Text einschränken.

Das Konsumieren setzt voraus, dass der Dokumentausschnitt in hinreichender Größe dargestellt wird. Dadurch erst ist es dem Benutzer möglich, das Dokument zu konsumieren, der Ausschnitt des Dokuments darf allerdings auch nicht zu kurz gewählt werden, weil sonst der Kontext völlig verloren ginge und der Inhalt nicht mehr sinnvoll wahrnehmbar wäre.

Suchen

Der Benutzer will eine Stelle im Dokument wieder zu finden, oder eine für ihn relevante Stelle des Dokuments zu finden.

Im ersten Fall versucht der Benutzer das erinnerte Bild der Stelle im Dokument wieder zu erkennen. Diese menschliche Fähigkeit setzt voraus, dass der wieder zu erkennende Inhalt bereits in ähnlicher Form rezipiert wurde[1].

Im zweiten Fall wird er nur als wichtig gekennzeichnete Teile des Dokuments (z.B.: Überschriften) bewusst wahrnehmen und an relevanten Stellen diese Wahrnehmung vertiefen, bzw. letztlich konsumieren.

SCROLLEN

Das Verschieben des dargestellten Dokumentausschnitts wird üblicherweise mit Scrollen (scrolling, panning) gelöst. Scrollen stellt in dieser Arbeit dadurch den Status Quo der Dokumentnavigation auf Desktopgeräten dar, dessen Schwächen hier zuerst dargelegt werden sollen, um Vergleichsmomente für die einzelnen Verbesserungsvorschläge zu erhalten. Scrollen ist eine Summe von Interaktionselementen und –modi, die offensichtliche (weil sichtbare) ist der Scrollbalken, die versteckte ist die Tastatur.

Scrollbalken

Ein Scrollbalken (scrollbar, Bildlaufleiste) besteht aus drei Elementen: der Lauffläche (through), dem Balken (bar, thumb, knob) und Schaltflächen mit Pfeilen (hier: „Pfeile“) an den Enden.

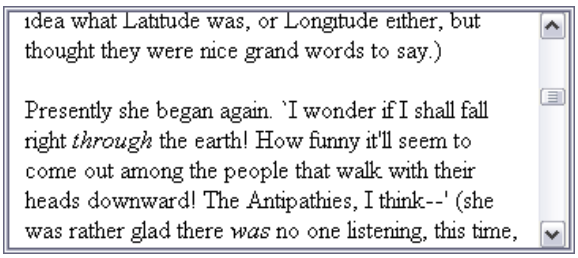

idea what Latitude was, or Longitude either, but
thought they were nice grand words to say.)
Presently she began again. 'I wonder if I shall fall
right *through* the earth! How funny it'll seem to
come out among the people that walk with their
heads downward! The Antipathies, I think--' (she
was rather glad there *was* no one listening, this time,

Abbildung 1: Scrollbalken bei langem Dokument[2]

Grundsätzlich unterscheiden wir hier drei Arten des Bedienens von Scrollbalken, die in gängigen Implementierungen (Abbildung 1 zeigt Scrollbalken in Windows) explizit visuell unterscheidbar sind:

- Positionieren des Balkens
- Blättern: Klicken auf die Lauffläche
- Zeile: Klicken der Pfeile

Diese drei Interaktionsmöglichkeiten versuchen dem Benutzer für die jeweilige Situation das richtige Werkzeug zur Verfügung zu stellen. Dennoch hat Scrollen wie es heute implementiert ist einige gravierende Nachteile, besonders wenn es auf kleinen Bildschirmen und/oder für sehr große Dokumente eingesetzt wird. Verstärkt werden diese Nachteile dadurch, dass zur direkten Interaktion mit Scrollbalken ein 2D-Eingabegerät erforderlich ist, was derzeit bei einem Großteil der mobilen Plattformen nicht vorausgesetzt werden kann.

[1] Das menschliche visuelle sensorische System erfasst aus einer Szene zuerst die informativsten Teile und benutzt periphere Bereiche der Netzhaut um den jeweils nächsten informativen Bereich zu finden. Diese Abfolge von Punkten bildet einen Erkennungspfad. Ähnlichkeiten in diesem Erkennungspfad resultieren im Wiedererkennen von Bildern [RGGPS98]. Dieses Modell impliziert eine grundsätzliche Abhängigkeit des Erkennens von der dargestellten Szene, das bedeutet es ist möglicherweise für diese Aufgabe nicht sinnvoll, dem Benutzer beliebige Ausschnitte von Dokumenten zu präsentieren. Page Flipping (siehe weiter unten) wäre in diesem Sinne besser für diese Benutzeraufgabe geeignet als andere vorgestellte Ansätze, weil es im Suchmodus immer dasselbe Bild je Seite zeigt.

[2] Als Beispieltext für die Abbildungen dient das erste Kapitel aus Alice’s Adventures in Wonderland [C65], dieses besteht aus ca. 2.200 Wörtern.

Tastatur

Die Tastatur dient in fensterorientierten Benutzerschnittstellen beim Scrollen meist nur als Ersatz für die Verwendung von Scrollbalken, ist allerdings das zentrale Eingabeelement für mobile Geräte.

Zum Scrollen werden die Pfeiltasten auf der Tastatur verwendet. Das Verhalten in typischer Dokumentnavigation ist analog zum Verwenden der Pfeilschaltflächen am Scrollbalken. Weiters kann auch das Klicken der Lauffläche durch die Bild-Auf (Page-Up) bzw. Bild-Ab (Page-Down) Tasten imitiert werden. Auf mobilen Endgeräten sind meist nur Pfeiltasten verfügbar.

Geschwindigkeitsbasiertes Scrollen

Geschwindigkeitsbasiertes Scrollen (rate based scrolling) benötigt ein eindimensionales Eingabegerät[3]. Die Grundidee ist, dass dieser Eingabeparameter die Scrollgeschwindigkeit bestimmt. So erlaubt diese Technik eine feine Abstufung zwischen zeilenweisem und seitenweisem Scrollen. Diese Art von Scrollen ist im Dokumentbrowsers oft verfügbar, wenn der Benutzer die mittlere Maustaste bzw. das Scrollrad klickt.

Geschwindigkeitsbasiertes Scrollen bietet eine breitere Palette an Anwendungsmöglichkeiten für Benutzeraufgaben, sobald jedoch eine gewisse Geschwindigkeit überschritten wird, kann es nicht mehr sinnvoll benutzt werden [WSC04]: Der Inhalt wird so schnell bewegt, dass er für den Betrachter nicht mehr erkennbar ist [IH00]. Meist wird dieser Effekt durch die Verwendung eines Bildschirms mit hoher Latenz (z.B. LCD) noch verstärkt.

Weitere Möglichkeiten

Es gibt weitere bekannte Implementierungen von Scrollen, Erwähnung soll hier das Verschieben und das mittlerweile weit verbreitete Scrollrad der Maus finden.

Beim Verschieben, kann das Dokument mit dem 2D-Eingabegerät im sichtbaren Ausschnitt verschoben werden. Dieser Modus findet zum Beispiel im Acrobat Reader Anwendung. Das Verschieben ist auch eine ideale Interaktionsmöglichkeit für Kartenmaterial oder ähnliche in zwei Dimensionen umfangreiche Dokumente.

Das Scrollrad (Mouse Wheel) bietet eine möglichst natürliche Interpretation des Scrollens. Analog zum Verschieben wird hier das Dokument im sichtbaren Ausschnitt in einer Dimension verschoben, aber ohne dabei die Maus zu bewegen.

Beide dieser Möglichkeiten werden im Vergleich berücksichtigt, aber jede Interaktionsmöglichkeit, die ein 2D-Eingabegerät erfordert, ist natürlich auf mobilen Geräten nur bedingt einsetzbar.

PROBLEME MIT SCROLLEN

Neben den grundsätzlichen Schwächen von Scrollen ergeben sich im Einsatz für Dokumentnavigation und auf mobilen Geräten mit kleinen Bildschirmen diverse Probleme für die Interaktion.

Orientierungslosigkeit und Kontextverlust

Die Benutzung von Schnittstellen zur Dokumentennavigation hinterlässt den Benutzer oft orientierungslos an einer Stelle des Dokuments. Dies tritt offensichtlich bei einem direkten Sprung im Dokument auf (beispielsweise durch einen Klick auf die Lauffläche in bestimmten Implementierungen von Scrollbalken), allerdings auch kontinuierliche Navigation (z. B.: Verwenden des Pfeil-Buttons an den Enden von Scrollbalken) verursacht selbige Orientierungslosigkeit, sobald eine etwas höhere Geschwindigkeit überschritten wird. Eine offensichtliche Lösung für dieses Problem wäre eine Beschränkung der Geschwindigkeit auf einen Maximalwert, bzw. beim direkten Sprung eine kontinuierliche Animation eines Scrollvorgangs mit selbiger Geschwindigkeit. Diese Maximalgeschwindigkeit würde zwar die temporale visuelle Kohärenz erhalten, bringt aber besonders bei relativ großen Dokumenten lange Interaktionszeiten mit sich, in denen der Benutzer aufmerksam das Geschehen am Bildschirm verfolgen muss, um nicht danach ebenso orientierungslos zu sein wie in der ursprünglichen Lösung.

Auswege aus diesem Dilemma müssen also höhere relative Navigations-Geschwindigkeiten erlauben und dennoch das geografische Orts-Bewusstsein des Benutzers im Dokument erhalten.

Ungenauigkeit

Die dem Benutzer überlassene Wahl des Navigationswerkzeuges beeinflusst sowohl die Performanz als auch die Genauigkeit des Navigationsvorgangs. Herkömmliche Werkzeuge bieten jeweils entweder Genauigkeit oder Geschwindigkeit, so muss der Benutzer oft eine Kombination von zwei Werkzeugen einsetzen, um die gegebene Aufgabe zu erfüllen, was die Initialkosten[4] der verwendeten Werkzeuge summiert.

Imperformanz

Wie bereits im obigen Absatz erklärt bedeutet Scrollen einen Trade- Off zwischen Genauigkeit und Performanz.

[3] Bekannte Implementationen von Geschwindigkeits-basiertem Scrollen greifen oft auf die y-Achse einer Maus als Eingabeparameter zurück.

[4] Die Initialkosten werden beispielsweise in der KLM-Analyse sichtbar, schließen aber auch Orientierungsphasen des Benutzers ein.

Einheitlichkeit
Die bisher verfügbaren Möglichkeiten des Scrollen stellen schon eine reichhaltige Auswahl an unterschiedlichen Werkzeugen dar. Diese Werkzeuge werden aber oft nicht vollständig genutzt, weil sie nicht konsistent über Plattformen und Anwendungen hinweg implementiert sind und so die Anwendung der Techniken nicht in den täglichen Gebrauch Eingang finden kann.

Mobile Geräte und kleine Bildschirme
Wir haben am Anfang dieses Kapitels darauf hingewiesen, dass das hier Beschriebene den Status Quo der Dokumentnavigation darstellt, doch auch diesse verbreiteten Techniken des Scrollen werden auf Mobilen Geräten oft nicht in vollem Umfang unterstützt. Durch die Verwendung von kleinen Bildschirmen wird das Dokument relativ gesehen noch größer, aber auch jedes weitere GUI-Element schränkt den Platz weiter drastisch ein. Weiters stellen mobile Geräte stellen auch meist keine Maus zur Verfügung. Wenn ein Stylus verfügbar ist, dann gilt zu bedenken, dass die Interaktion bestimmte Teile des GUI überdeckt. Durch diese Einschränkungen und den schnellen Generationswechsel in diesem Bereich sind Hersteller von Mobiltelefonen besonders experimentierfreudig wenn es um den Einsatz von neuen Techniken geht.

LÖSUNGSANSÄTZE

Im Folgenden wird eine Auswahl von Lösungen, bzw. Lösungsstrategien vorgestellt, die oben beschriebene Schwächen adressieren, jede Lösung wird bezüglich ihrer Durchführbarkeit auf mobilen Geräten diskutiert. Zu den aufgezählten Lösungsansätzen werden, soweit vorhanden und sinnvoll, Beispiele präsentiert.

Alternative Sensoren
Um die bereits erwähnten Einschränkungen von mobilen Geräten zu überwinden, könnten bisher domainfremde Sensoren zum Einsatz kommen. Ein einfaches Beispiel ist das Scrollrad am Mobiltelefon, das bereits in einigen Geräten zu finden ist, aber auch Bewegungssensoren (Accelerometer) und Neigungssensoren [LKM07] oder auch Positionssensoren könnten sich gut für die Anwendung eignen.

Jede neue Sensortechnologie birgt aber neben ihren Vorteilen möglicherweise auch Nachteile, so können beispielsweise Neigungssensoren durch die notwendige flexible Neigung des Geräts bei bestimmten Winkeln, Lichteinfall und Spiegelungen die Lesbarkeit des Dokuments beeinträchtigen [EM04].

Verschieben des Fensters anstatt des Inhalts
Diese Idee ist ein konzeptueller Lösungsansatz, erfordert aber auch die Verwendung von neuen Sensoren. Die Position im virtuellen Dokument könnte an die reale Welt gebunden sein und so ist der Bildschirm eine Art „Magische Lupe“[5] durch die das Dokument sichtbar wird. Zwar ist der Kontext nicht explizit sichtbar und das Suchen im Dokument ist wie beim Scrollen noch immer vergleichbar mit dem Verwenden einer Taschenlampe in einem dunklen Raum, allerdings nutzt diese direktere Umsetzung des Scrollens das räumliche Vorstellungsvermögen des Menschen [Y03] und könnte möglicherweise auch den kinästhetischen Sinn des Menschen nutzen.

Scrollen mit eingebauter Kamera
Viele mobile Geräte sind bereits mit einer Kamera ausgestattet, die auch zur Navigation benutzt werden könnte. So könnte das „Verschieben des Fensters anstatt des Inhalts“ mit dieser Technik einfach umgesetzt werden, eventuell sogar erweitert um Elemente außerhalb des mobilen Geräts, so könnte der Kontext[6] durch den Benutzer mit Repräsentationen von Inhaltsstellen angereichert werden[7].

Dieses direkte Mapping von Hintergrund zu virtuellem Inhalt kann aber auch aufgeweicht werden[8], und so können die 2D-Positionsdaten zu jeder beliebigen Art von 2D-Navigation eingesetzt werden. Ein offensichtlicher Nachteil beim Einsatz dieser Technik für lange Dokumente ist, dass der Benutzer in der Bewegung des Geräts an physische Grenzen stoßen kann [HMSC05].

Der gegenläufige Ansatz wäre, ein 2D-Eingabegerät analog zu einer Maus mit dieser Technik umzusetzen.

Overview and Detail
Eine Navigationsansicht (overview) ist ein (meist kleines) Fenster, das das gesamte Dokument in einer Überblicksansicht zeigt. Der aktuell im Hauptfenster sichtbare Ausschnitt wird mittels eines Rahmens in diesem Fenster angezeigt. Dieser Rahmen kann vom Benutzer verschoben werden, wodurch sich die Anzeige im Hauptfenster entsprechend ändert.

Der Benutzer muss zur Verwendung dieses Systems sein Blickfeld konstant zwischen den zwei Ausschnittsfenstern

[5] Der Begriff „magic lens“ wurde im Xerox PARC für bewegliche visuelle Darstellungsfilter für Daten geprägt.

[6] In diesem Fall könnte der Kontext zum Beispiel ein Schreibtisch sein.

[7] Dies setzt allerdings voraus, dass sich der Benutzer in derselben Umgebung befinden muss, um von dieser externen Kontextinformation zu profitieren, eine Voraussetzung die gerade für mobile Geräte möglicherweise zu streng ist.

[8] Möglicherweise verursacht diese losere Kopplung eine Reduktion im Nutzungsgrad von räumlichen Vorstellungsvermögen und kinästhetischem Sinn.

wechseln, was bei jedem Wechsel mit einer Orientierungsphase verbunden ist.

Vergleichstests zeigen bessere Leistungswerte für dieses Interface auf kleinen Bildschirmen als für Scrollen [GF04], für die Anwendung der Navigation in großen Dokumenten könnte das Übersichtsfenster das gesamte Dokument als Streifen links oder rechts vom vergrößerten Ausschnitt angezeigt werden.

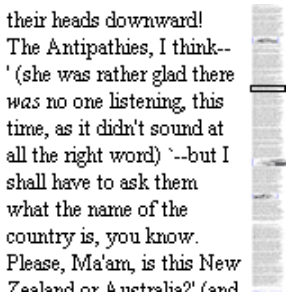

their heads downward!
The Antipathies, I think--
' (she was rather glad there
was no one listening, this
time, as it didn't sound at
all the right word) `--but I
shall have to ask them
what the name of the
country is, you know.
Please, Ma'am, is this New

Abbildung 2: Overview and Detail

country is, you know.
Please, Ma'am, is this New
Zealand or Australia?' (and
she tried to curtsey as she
spoke--fancy *curtseying* as
you're falling through the

Abbildung 3: Focus and Context

Focus and Context

Dieser Ansatz vereint die zwei Ansichtsgrößen der „Overview and Detail“ Technik in einer Darstellung, indem das Detail unverzerrt als Focus dargestellt wird, während der gesamte Kontext in verzerrter Form am Rand des Bildschirms angezeigt wird.

Eine Implementierung dieser Technik sind Fisheye-Views, die auf kleinen Bildschirmen trotz der durch die geringe Auflösung starke Verzerrung des Inhalts in Benutzertests im Vergleich zum traditionellen Scrollen gut abscheiden [GF04]. Zur Dokumentnavigation eignet sich dieser Ansatz allerdings nicht gut, da es aufgrund der Dokumentgröße nicht sinnvoll ist, den gesamten Kontext im peripheren Bildschirmbereich darzustellen.

Zooming Interfaces

Zoomen stellt in diesem Kontext eine Analogie zum Scrollen dar: anstatt nur das Dokument unter dem Ausschnitt zu verschieben, kann zusätzlich eine Vergrößerung (oder, je nach Anwendungsfall Verkleinerung) bestimmt werden. So kann der Benutzer eine Übersicht eines großen Teils des Dokuments erlangen, um von dort aus wieder einen bestimmten Teil des Dokuments zu vergrößern. Zooming Interfaces [BH94] (ZUIs) gehen noch einen Schritt weiter: sie ersetzen ganze Teile von Dokumenten durch Symbolische Platzhalter, die im jeweiligen Vergrößerungsgrad noch lesbar sind.

Zooming Interfaces sind viel weiter gefasst, eignen sich jedoch auch hervorragend zum Navigieren von Dokumenten.

Pad++, eine Implementierung von Zooming Interfaces verwendet zwei von drei Maustasten zum Zoomen, eine zoomt heraus, die andere hinein. Die Zoomoperation hat den Mauszeiger als Zentrum, dadurch kann der Benutzer während des Zoomvorgangs denselben beeinflussen [BH94]. In dieser Implementierung ist kein vereinfachter Lesemodus vorgesehen, allerdings wäre dieser einfach mit einer dritten Taste realisierbar, mit der das Verschieben realisiert wird.

Diese drei explizit notwendigen Interaktionsmodi machen das Arbeiten mit diesem System allerdings weniger intuitiv und auf einem mobilen Gerät nahezu unmöglich, da selbst wenn das benötigte 2D-Eingabegerät mit einem Stylus implementiert ist, die nötige Wahl der Modi schwer fallen wird.

Speed Dependent Automatic Zooming

Ein Ansatz zur Lösung ist Speed Dependent Automatic Zooming, das durch verkleinerte Darstellung des Dokuments während des Navigationsvorgangs den Kontext des aktuellen Ausschnitts explizit darstellt.

Ein eindimensionaler[9] Eingabeparameter bestimmt die Geschwindigkeit des Scrollvorgangs[10], sobald dieser Parameter einen gewissen Wert überschreitet wird der Zoomfaktor berechnet und das Dokument entsprechend kleiner dargestellt. Der aktuelle „virtuelle“ Ausschnitt wird mit einem Rahmen über der verkleinerten Ansicht dargestellt [IH00]. Typischerweise ist die Geschwindigkeit nach oben limitiert, um die Ansicht des Dokuments nicht zu einem schmalen Streifen degenerieren zu lassen.

Dieser Ansatz bietet viele der Vorteile von Zooming Interfaces, eliminiert allerdings deren gravierenden Nachteil, indem er das Zoomen als implizite Aktion beim Scrollen realisiert.

SDAZ hat aber noch großes Optimierungspotenzial, so werden komplexere Berechnungen der Scrollgeschwindigkeit und des Zoomfaktors im Verhältnis zum Eingabewert erforscht, wie auch dynamisches Eingabeverhalten [EM04].

Trotz der eher exotischen und unbekannten Umsetzung verliefen erste Benutzertests im Anwendungsbereich der Dokumentnavigation sehr positiv und erhielten gute Akzeptanznoten verglichen mit reinem Scrollen [EM04, IH00].

Diese Technik eignet sich besonders gut, um mit verschiedenen Eingabesensoren kombiniert zu werden, es wird zum Beispiel die Verwendung von Neigungssensoren erforscht [EM04].

Page Flipping

Page Flipping versucht die Navigation in Papierdokumenten möglichst effizient nachzubilden und kombiniert dabei mehrere Navigationsmethoden, aber wie

[9] Diese Technik wäre einfach um eine Dimension erweiterbar.

[10] vergleiche geschwindigkeitsorientiertes Scrollen

schon SDAZ verwendet es nur einen Eingabeparameter. Es bietet mehrere Interaktionsphasen, bei niedrigen Werten des Eingabeparameters setzt es SDAZ ein, sobald ein bestimmter Grenzwert überschritten ist, wird ein *flipping mode* umgeschaltet, der die Seiten des Dokuments gesamt und einzeln am Bildschirm darstellt. Sobald der Benutzer die gewünschte Seite erreicht, senkt er den Eingabewert abrupt auf 0 und aktiviert so die letzte Navigationsphase, das *backtracking*. In dieser Phase wird die Endposition des Vorgangs aus der Geschwindigkeit und der typischen Reaktionszeit des Menschen berechnet und angezeigt [SG05].

Benutzertests zeigen für diese Technik deutlich bessere Effizienzwerte als für Scrollen aber auch besser als für herkömmliches SDAZ, welches in den Tests dieser Forschungsgruppe etwas schlechter abschneidet als herkömmliches Scrollen [SG05].

Die Umsetzbarkeit dieser Technik ist mit SDAZ vergleichbar, weil dieselben Eingabetechniken verwendet werden, hier wird aber ein selbst zentrierendes Eingabegerät bevorzugt, um den abrupten Wertabfall sinnvoll realisieren zu können.

Verbesserung der Bildschirm- Technologie

Das Lesen digitaler Dokumente auf mobilen Geräten könnte natürlich auch durch eine Verbesserung der Darstellungsgeräte angenehmer und effizienter gestaltet werden. Ein offensichtlicher Angriffspunkt ist die Auflösung und die Größe dieser Bildschirme, die sicherlich noch einiges Verbesserungspotential haben. Jedoch ist die Größe des Bildschirms limitiert durch die maximal sinnvolle Größe von mobilen Geräten und die Auflösung durch die maximal sinnvoll wahrnehmbare Auflösung.

Diesen Einschränkungen gegenüber stehen aber kreative Lösungsansätze wie die Verwendung von zwei Bildschirmen auf gegenüberliegenden Flächen des mobilen Geräts, die durch Wenden des Geräts in verschiede Richtungen intuitive Navigationsmöglichkeiten eröffnen [N05]. Diese Interaktionsmöglichkeit erfordert zur effizienten Verwendung die Darstellung eines Ausschnitts sinnvoller Größe und ist somit mit derzeitiger Bildschirmtechnologie nur eingeschränkt verwendbar.

VERGLEICH

Anschließend wird ein Vergleich zwischen den beschriebenen Techniken durchgeführt (siehe Tabelle 1, folgende Seite). In der Literatur sind nicht für jede Technik alle hier bewerteten Eigenschaften ausgeführt, bzw. auch mit empirischen Werten untermauert, an jenen Stellen finden sich in der Vergleichsmatrix Schätzungen.

Eine KLM[11]-Analyse bietet hier keine gute Bewertungsbasis, nachdem bei vielen Techniken bestimmte zeitliche Abläufe der Mausbewegung erforderlich sind und so die Schätzwerte für die Positionierungs-Aktion (P) wohl unpassend sind. Diese dynamische Steuerung bewirkt auch ständige Kognitive Leistung (M) während des Vorgangs. Deshalb wird der Aufwand der Interaktion hier in die oben beschriebenen zwei Anwendungsfälle getrennt bewertet.

SDAZ ist und dessen Erweiterung Page Flipping schneiden in diesem Vergleich aufbauend auf den guten Testergebnissen in der Literatur als deutliche Gewinner des Vergleichs ab, doch auch bei älteren Lösungsansätzen wie Focus and Context drängt sich die Frage auf, warum diese Techniken bisher noch keine breite Anwendung in Benutzerschnittstellen gefunden haben.

FAZIT

Trotz des langen Bestehens und der Allgegenwärtigkeit von Scrolltechniken bieten viele der vorgestellten Alternativen signifikante Vorteile im Vergleich zu traditionellen Techniken, besonders für die Anwendung auf mobilen Geräten. Der bisherige Zugang, für verschiedene Aufgaben verschiedene Werkzeuge anzubieten, der für mobile Geräte aufgrund der limitierten Eingabemöglichkeiten ohnehin nur begrenzt anwendbar ist, wird vielleicht in Zukunft abgelöst von einheitlichen Lösungen, die in allen Anwendungs-Situationen akzeptable Leistung bieten.

Generalisierbarkeit der Techniken

Die vorgestellten Techniken sind natürlich durchwegs das Resultat von Forschung in bestimmten Gebieten, doch wie in dieser Arbeit Techniken vorgestellt wurden, die ursprünglich nicht zur Dokumentnavigation entwickelt wurden, können wir feststellen, dass deren Anwendung nicht auf diese Gebiete beschränkt bleiben muss.

Es bieten sich je nach Technik verschiedene Arten der Generalisierbarkeit, so könnte mit den beschriebenen neuen Sensoren etwa ein Mauszeiger gesteuert werden, wodurch jede bekannte Interaktionsmöglichkeit mit 2D-Eingabegeräten nachgebildet werden kann. Aber auch jene Techniken die die Darstellung des Inhalts und das Verschieben des Ausschnitts adressieren können ebenso oder in abgewandelter Form für die Navigation in Kartenmaterial, Bildergalerien, Objektlisten, Filmmaterial oder großen Benutzerschnittstellen Einsatz finden.

Ausblick

Speed Dependent Automatic Zooming und dessen Varianten zeichnen sich durch gute Werte bei Benutzertests aus und werden wohl unausweichlich in naher Zukunft für die Dokumentnavigation auf kleinen Bildschirmen Einsatz finden, doch auch die Entwicklung neuer Hardware für Bildschirm- und Sensortechnologie verspricht Besserung der derzeitigen Situation. Es bleibt zu hoffen, dass sich neue Sensoren auf mobilen Plattformen etablieren, die zumindest eindimensionale Benutzereingabe für die vorgestellten Techniken erlauben.

[11] Keystroke-Level Model (KLM) ist ein Analysemodell für Mensch-Computer Interaktion das auf GOMS basiert.

Sobald die notwendigen technischen Rahmenbedingungen ausreichende Verbreitung erreicht haben, ist zu erwarten, dass die Mobiltelefonbranche einige dieser neuen Techniken adaptieren wird, schon alleine um Kaufanreize für neue Generationen von Geräten zu schaffen, die trotz der bereits hohen Verbreitungsdichte die Umsätze der Hersteller erhalten sollen.

Lösung	Orientierung	Kontext	Akzeptanz	Genauigkeit Lesen	Genauigkeit Sprung	Performanz Lesen	Performanz Sprung	Durchführ-barkeit auf mobilen Geräten
Scrollen: Positionieren	-	0	+	0	0	-	+	--
Scrollen: Blättern	+	0	0	--	+	+	0	+
Scrollen: Zeile	++	+	++	+	0	0	--	++
ZUI	++	+	+	+	+	+	-	--
SDAZ	++	++	+	++	++	+	+	0[12]
SDAZ + Tilt	+	+	0	++	+	+	+	+
Kamera	+	0[13]	0	+	0	+	+	++
Scroll-Wheel	++	0	++	+	0	++	--	+
Overview & Detail	+	+	-	--	+	-	+	-
Focus & Context	++	++	+	+	+	+	+	-
Page Flipping	+	++	+	++	++	+	++	0

REFERENZEN

[**IH00**] Takeo Igarashi, Ken Hinckley, Speed-dependent automatic zooming for browsing large documents, in: ACM Press, Proceedings of the 13th annual ACM symposium on User interface software and technology, ACM Press New York, USA, 2000, p139-148

[**CSW05**] Andy Cockburn, Joshua Savage, Andrew Wallace, Tuning and testing scrolling interfaces that automatically zoom, in: ACM Press, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press New York, USA, 2005, p71-80

[**EM04**] Parisa Eslambolchilar, Roderick Murray-Smith, Tilt-Based Automatic Zooming and Scaling in Mobile Devices, in: Springer, Proceedings of Mobile HCI 2004 (Glasgow), Springer, 2004

[**LKM07**] Hyun-Jean Lee, Madhur Khandelwal, Ali Mazalek, Tilting table: a movable screen, in: ACM Press, Proceedings of the 1st international conference on Tangible and embedded interaction, ACM Press New York, USA, 2007, p93-96

[**HBP02**] Kasper Hornbæk, Benjamin B. Bederson, Catherine Plaisant, Navigation patterns and usability of zoomable user interfaces with and without an overview, in: ACM Press, Source ACM Transactions on Computer-Human Interaction (TOCHI) Volume 9, Issue 4, ACM Press New York, USA, 2002, p362-389

[**WSC04**] Andrew Wallace, Joshua Savage, Andy Cockburn, Rapid visual flow: how fast is too fast?, in: ACM Press, Proceedings of the fifth conference on Australasian user interface - Volume 28, ACM Press New York, USA, 2004, p117-122

[**SG05**] Liyang Sun, François Guimbretière, Flipper: a new method of digital document navigation, in: ACM Press, CHI '05 extended abstracts on Human factors in computing systems, ACM Press New York, USA, 2005, p2001-2004

[12] Mobile Geräte bieten zugunsten der Stromaufnahme limitierte Rechenleistung, werden aber immer leistungsfähiger wodurch in Zukunft die Durchführbarkeit von Real-Time-Grafik auf mobilen Geräten steigen wird [vgl. MNC07].

[13] Wie weiter oben angedeutet birgt diese Lösung in bestimmten Implementationen Potenzial in dieser Bewertungskategorie.

[14] Bewertungen: sehr schlecht (--), schlecht (-), neutral (0), gut (+), sehr gut (++)

[**HMSC05**] Antonio Haro, Koichi Mori, Vidya Setlur, Tolga Capin, Mobile camera-based adaptive viewing, in: ACM Press, Proceedings of the 4th international conference on Mobile and ubiquitous multimedia, ACM Press New York, USA, 2005, p78-83

[**Y03**] Ka-Ping Yee, Peephole displays: pen interaction on spatially aware handheld computers, in: ACM Press, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press New York, USA, 2003, p1-8

[**N05**] Nakamura, S. 2005. Reversible display: content browsing with reverse operations in mobile computing environments. In *Proceedings of the 7th international Conference on Human Computer interaction with Mobile Devices &Amp; Services* (Salzburg, Austria, September 19 - 22, 2005). MobileHCI '05, vol. 111. ACM Press, New York, NY, 339-340. DOI= http://doi.acm.org/10.1145/1085777.1085855

[**GF04**] Gutwin, C. and Fedak, C. 2004. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. In *Proceedings of the 2004 Conference on Graphics interface* (London, Ontario, Canada, May 17 - 19, 2004). ACM International Conference Proceeding Series, vol. 62. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, 145-152.

Weiterführende Literatur

[**BH94**] Benjamin B. Bederson, James D. Hollan, Pad++: a zooming graphical interface for exploring alternate interface physics, in: ACM Press, Proceedings of the 7th annual ACM symposium on User interface software and technology, ACM Press New York, USA, 1994, p17-26

[**BGBS02**] Patrick Baudisch, Nathaniel Good, Victoria Bellotti, Pamela Schraedley, Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming, in: ACM Press, Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, ACM Press New York, USA, 2002, p259-266

[**A05**] Tue Haste Andersen, A simple movement time model for scrolling, in: ACM Press, CHI '05 extended abstracts on Human factors in computing systems, ACM Press New York, USA, 2005, p1180-1183

[**PMJJ06**] Dynal Patel, Gary Marsden, Matt Jones, Steve Jones, Improving photo searching interfaces for small-screen mobile computers, in: ACM Press, Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, ACM Press New York, USA, 2006, p149-156

[**RGGPS98**] Rybak, I. A., Gusakova, V. I., Golovan, A. V., Podladchikova, L. N., and Shevtsova, N. A. (1998) A model of attention-guided visual perception and recognition, in: Vision Research 38, Elsevier Science Ltd., 1998, p2387-2400

[**MNC07**] Mulloni, A., Nadalutti, D., and Chittaro, L. 2007. Interactive walkthrough of large 3D models of buildings on mobile devices. In Proceedings of the Twelfth international Conference on 3D Web Technology (Perugia, Italy, April 15 - 18, 2007). Web3D '07. ACM Press, New York, NY, 17-25. DOI= http://doi.acm.org/10.1145/1229390.1229393

[**C65**] Lewis Carroll, Alice's Adventures in Wonderland, 1865, 1991, Macmillan, online unter http://www.cs.cmu.edu/~rgs/alice-I.html, 27. Mai 2007

Using semantically enriched information and metaphors to visualize knowledge

Dipl. Ing. (FH) Alexander Abl
University of Klagenfurt
Universitätsstraße 56, 9020 Klagenfurt, Austria
Alexander.abl@itec.uni-klu.ac.at
Version 1.4(release)

ABSTRACT
Knowledge visualization using semantically enriched information is the next step in giving computer based systems the ability to do a job that has had to be done by humans all the years ago – to prepare information in a way that it could be understood without specialized domain know how. In Berners-Lee vision of the Semantic Web [2], agents need to understand the information on the web to accomplish their mission, which is to offer knowledge humans would have needed hours to search for. But semantics could be used as well, to build visualizations that humans could easily understand, even without domain knowledge. This article gives an introduction about the idea and shows that metaphors are one approach to convoy knowledge.

Author Keywords
Semantic Web, knowledge, visualization, ontology, metaphor.

INTRODUCTION
Vannevar Bush has already stated in his article "As we may think" [7] in the year 1945 that one major problem, humans will face in the next century, is the processing and conveyance of information and knowledge. Nowadays an incredible huge amount of information is stored on computers world wide and the technologies, as we use it today, encounter more and more difficulties to extract what we need. Most of the information out there is hardly processable automatically. If you look for something at Google you may get what you want but more likely you may get what you don't want. Some years ago researchers started to think about this problem, like the ideas presented in [6] to handle big digital libraries visually.

The *Semantic Web* which has been introduced by Tim Berners-Lee in [2] may be the successor of the web as we know it today. It offers some very interesting research areas, especially the visualization of semantic relationships between pieces of information to present knowledge. Knowledge processing and transformation are some key issues to handle the amount of data available today.

To be more precise, the *Semantic Web* offers, beside the human readable information, a machine interpretable form. The difference between information (a telephone number) and knowledge (like to know how to ride a bicycle) is the way how to transfer it to humans minds [3]. The transfer of information needs only the transportation of facts representing this information. To transfer knowledge some more sophisticated processes are necessary because it needs to be *re-created* in the target mind. The following sections introduce the idea of using semantically enriched data and metaphors for knowledge visualization and explain some important terms in this context like *ontology* and *taxonomy*. It presents how these techniques could be used to visually present information in an efficient way to derive some knowledge out of it. Further, it is described how techniques like Shneiderman's mantra *Overview first, zoom and filter, than details on demand* [8] and the tools introduced in [5] to visualize semantics enriched with metaphors like the in [3] introduced tube map can be used for that purpose. The major aspect of this article is to introduce the process of creating a knowledge visualization out of the information to be transferred and the domain know how held by domain experts. It also should be an anchor for further studies in this field.

KNOWLEDGE TRANSPORTATION
The transportation of knowledge is a main process in knowledge management today. Knowledge available in organizations is mostly expressed in text and numbers und must be interpreted by a human recipient who has to be a domain expert in most cases. This scenario is applicable on full scale to the information technology (IT) world. In contrast other businesses already use different approaches: an architect draws plans and builds models of the final work; engineers in mechanics use drawings as well to design parts of something and even in fields where visualizations are not common, they are used more and more like in chemistry (atom models) or medicine (spatial modelling of the human body). To use visualizations in the knowledge transfer was an idea of Burkhard [4]. He has discovered that in knowledge management the aspect of visualization to transfer knowledge has not been considered, which is an antagonism to the humans strongest and most important sense – the visual recognition.

The idea of visualizing information is old, almost as old as mathematicians have used diagrams to give their figures a

meaning. Diagrams may be used for applications with a mathematical background and mostly they are only interpretable in a specified context (someone has to know the calculations behind). The knowledge available in organizations is in most cases not describable by mathematical methods and therefore techniques used to display mathematical results and relationships are not applicable. Examples are processes in project management, knowledge about systems and applications and knowledge stored in libraries. Even if this knowledge is written down somewhere, if it should be transferred between human minds, it starts to get difficult. Depending on the complexity of certain knowledge the *target mind* has to learn something about the domain to be able to interpret the offered information as domain specific knowledge and to get the ability to make inferences. In most cases it would be an advantage for someone to know only the really necessary; it would be easier if some knowledge could be transferred somehow without knowing the context. This aspect becomes more important as our businesses need fast and precise decisions which cannot be carried out in a timely fashion if decision takers must be experts everywhere (e.g. executive summaries). To do that, the rules and relations human brains apply to information (if they know the domain) must be simulated by some computer based system. These rules and relations are called *semantics* in the information science and will be explained in the next section. These semantics are the starting point in the process of knowledge visualization.

SEMANTICS, ONTOLOGIES AND TAXONOMIES

Semantic is information about information, it is a kind of meta information that tells the reader something about the meaning, the context or the source of a piece of information. After [2] has been published, a process started within several organizations[1] to develop a framework showing how to apply semantics to information. These developments ended up in a still ongoing standardization process by the W3C[2] to introduce a language to describe semantics: the Web Ontology Language (OWL) [23]. But what is an *ontology*? There is not only one definition for the term *ontology* but several. One could be: ”An ontology defines the common words and concepts (the meaning) used to describe and represent an area of knowledge.”[9] or the first, short and precise definition by Gruber: ”An ontology is an explicit specification of a conceptualization.”[10]. It has to be said that this term has been borrowed from philosophy where it is defined as “...a branch of metaphysics concerned with the nature and relations of being.” [11]. In the information science it specifies a set of rules and relations which can be applied to information existing in the specified domain (as already mentioned above).

	Systematic of a lion
Regnum	*Animalia*
Classis	*Mammalia*
Ordo	*Ferae*
Genus	*Felis*
Species	*Felis leo*

Table 1: Carl von Linné classification of a lion according to Systema Naturae (1760)

The term taxonomy has been taken from the biological science and from librarians; both areas where things have to be classified. Again the general definition is: “The study of the general principles of scientific classification: systematics or classification; especially: orderly classification of plants and animals according to their presumed natural relationships.” [11]. Doconta’s explanation of taxonomy is as follows: “...a way of classifying or categorizing a set of things - specifically, a classification in the form of a *hierarchy*. A *hierarchy* is simply a treelike structure. Like a *tree*, it has a *root* and *branches*. Each branching point is called a *node*.” [9].

To give an example Table 1 shows an early taxonomy of animals on the example of a lion. At the times of Carl von Linné this was a very important step. Going up the taxonomy from the bottom to the top the terms become more general; going down from the top to the bottom the terms become more specialized. Therefore such taxonomies are also called generalization/specialization taxonomies.

Taxonomies already carry some meta information which can be used to describe an object semantically. From the example above it can be seen that, e.g. the species lion is in the class of mammals and must therefore be a mammal. It can also be inferred what e.g. a human has in common with a lion: both are mammals. Taxonomies are much weaker and simpler than ontologies. With ontologies more complex semantic relationships can be expressed. Ontologies can do a lot more than classification: they can express complex relationships among classes of objects or instances of these classes.

In [12] an example for a simple ontology is given as shown in Figure 1[3]. Already this simple example shows that with ontologies more complex semantics can be expressed. The picture shows inheritance relations like that a *NamedPizza* is a subclass of *Pizza* or a *DeepPanBase* is a subclass of

[1] OIL: The Ontology Inference Layer see http://www.ontoknowledge.org/oil/ and DAML: DARPA Agent Markup Language see http://www.daml.org/ which ended up in DAML+OIL the predecessor of OWL.

[2] The World Wide Web Consortium (http://www.w3c.org/)

[3] The ontology drawing has been created with the Protègè ontology editor (http://protege.stanford.edu/)

PizzaBase. Inheritance is expressed wit the *isa* relation. An *Isa* relations could be expressed with taxonomies as well but relations, as indicated by the dashed line, needs some extended mechanism. In Figure 1 the *isBase* relation indicates that every *Pizza* has a base or vice versa that a *PizzaBase* is always a part of a *Pizza*. The same applies to the *PizzaTopping* – every Pizza has one (the inverse is not drawn because of the little space).

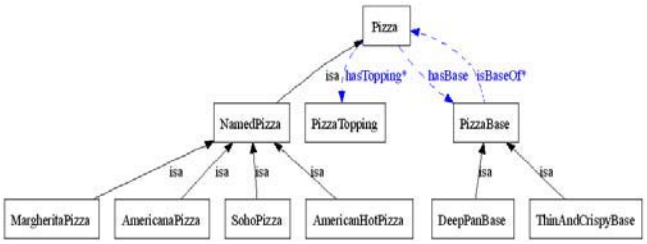


Figure 1: The Pizza ontology [12]

METAPHORS MEETS SEMANTICS

Humans have impressive abilities to absorb and compute visually provided information very efficiently. This topic is investigated beside the IT in disciplines like the psychology of learning, the cognitive psychology or in visual neuroscience. The advantages gained by using results from these science fields are listed in [9] and are as follows:

- visualizations trigger emotions (e.g. advertisements),
- express relationships (e.g. diagrams),
- show exceptions, trends or patterns,
- help to keep attention,
- help to keep things in memory,
- show an overview and details simultaneously (e.g. maps),
- make learning easier,
- help to coordinate a group of people (e.g. tactical drawings in team games) and
- motivate and activate the human spirit for new ideas.

In some of the above mentioned situations metaphors can be employed profitable. Metaphors support in keeping things in memory (mnemonics) and make it easier to learn. These two items are the key issues in knowledge transportation.

Important science disciplines

In the literature the science fields are somehow differentiated from each other. To give the reader the ability to categorize the type of science, this article deals with, a short classification is given here. Beside the science of visual perception and cognition which deals with the human abilities as mentioned above there are the

- Visual media science as defined in [13] which deals with science of pictures from ancient times till present,
- Information design as defined by Horn in [14] as "…the art and science of preparing information so that it can be used by human beings with efficiency and effectiveness."
- Information visualization which is a scientific discipline that deals with the visualization of huge digital libraries with interactive methods. Examples are given in [6], [8] and [1] and
- Knowledge visualization as defined in [3] deals with the finding of the best visualization method for a given situation to support the knowledge transfer and the process of generating new knowledge out of already available.

Metaphors in interactive systems

Metaphors are quite old which is already indicated by the word metaphor itself which is originated in the ancient Greek language with the meaning of "…carrying something somewhere else". A metaphor shows the way from something known to something new and unknown. Metaphors have been used in supporting the human learning process since centuries.

Metaphors are some essential element in the design of interactive systems. Again the definition of the term metaphor from the Merriam-Webster dictionary is as follows: "…a figure of speech in which a word or phrase literally denoting one kind of object or idea is used in place of another to suggest a likeness or analogy between them" [11]. This term is applicable as well to the visual media science and visual metaphors are used in the computer science as long people deal with graphical user interfaces.

Figure 2: Frequently used metaphors in interactive systems

In Figure 2 two frequently used metaphors are shown, the *Recycle Bin* on the left represents a metaphor from real life as an association where to but waste; the *key* on the right side represents a metaphor in the context of security in an IT based system. According to Eppler [18], metaphors are based on the following phenomena, objects, activities and concepts:

- Natural phenomena: e.g. iceberg, mountains, waterfall, tree, abyss, volcano, river, etc.
- Man-made objects: e.g. bridge, funnel, road, balance ladder, castles, stairs, umbrella, bucket, pendulum, lever, etc.
- Activities: e.g. climbing, walking, driving, eating, fishing, hunting, fencing, etc.

- Concepts: e.g. family, friends, job, war, law, chaos, peace, etc.

The example in Figure 3 shows the metaphor of a bridge showing how to successful lead negotiations based on the Harvard[4] negotiation method. The bridge is a kind of metaphor based on the man-made objects. It helps the viewer to memorize the knowledge provided within the picture more easily and effectively.

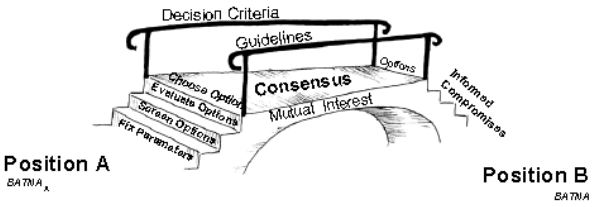


Figure 3: Metaphor of a bridge taken form [18]

Current usage of semantics and metaphors

In the Internet there are already examples how information can be augmented with semantics and then displayed using a kind of metaphors. The French search engine KartOO [15] uses the metaphor of smaller and bigger documents to visualize the most appropriate source for the search keyword.

Figure 4: The *semantic* search engine KartOO [15]

A typical search result of KartOO is shown in Figure 4. In the middle of the screen a map is shown displaying found documents and semantic links that represent similar keywords found in both documents. KartOO does not search for semantics stored in web documents, because nowadays there is mostly no semantic stored there, but it tries to find similarities in the search result to generate a very simple kind of semantic. This is then used to generate a graphical interface using the metaphor of interlinked documents.

[4] Found in [18] where it is referenced to: Lewicki, R.J., Saunders, D. M., Minton, J. W. (1997) "Essentials of Negotiation." Mcgraw-Hill Professional

Another approach by the same authors is Ujiko [16] which can be seen as refinement of KartOO using the lessons learned there and it is based on an interesting idea: the user gets a bonus point for every click on a search result s/he does. Every time when reaching a threshold s/he gets a kind of reward in form of additional functions, new skins or more fancy colours. The metaphor used for Ujiko is a multimedia player, displaying search results as play-lists and using colour indications to show the appropriateness of a result. Ujiko does not use semantics provided on web pages (it faces the same problem as KartOO) but it uses the ontologies in the users brain to link keywords to search results. This helps to gain the ability to retrieve better search results per keyword.

Figure 5: The refinement of KartOO, the search engine Ujiko [16]

Another interesting approach, similar to those of KartOO and Ujiko is the Grooker [19] search engine. Grooker is a kind of collection overview which can apply filters and zooming on demand.

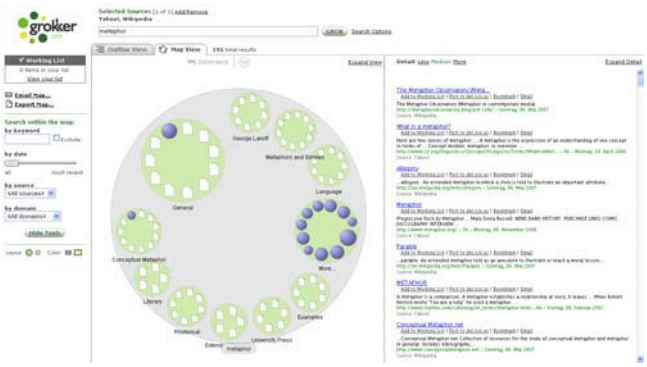


Figure 6: Grooker uses the details on demand paradigm and allows to apply filters [19]

How Grooker looks like is shown in Figure 6:. The circle in the middle represents the searched domain, on the left there are some widgets for an easy application of filters (keywords, time slider, etc.) and on the right there is the result set displayed (list view). The map itself is interactive what means that by clicking in one of the smaller circles (every circle represents a classification) it expands and allows dragging deeper and deeper. The source Grooker uses is either Yahoo or Wikipedia or both. To gain some semantics Grooker uses a gaming approach. Interested users

can play a kind of puzzle to annotate terms with semantics while playing. Grooker is intended to be a search engine within organizational intranets because its inventors believe that the content of all company databases is about 500 times larger that the public web.

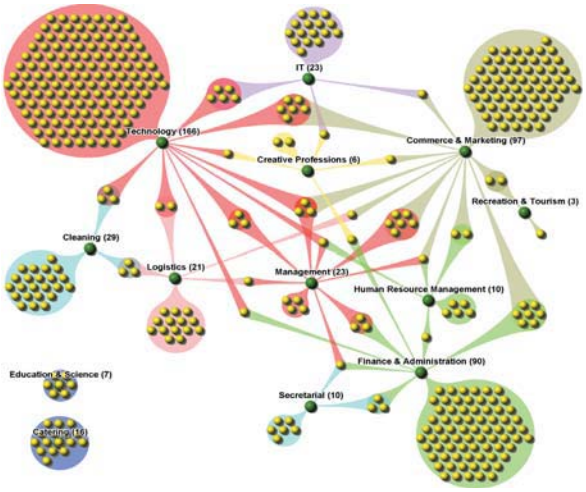


Figure 7: Spectacles cluster map [5]

In [5] the software Spectacle[5] is introduced. Spectacle is a tool to search databases and visualize the result as a map. As semantic a simple taxonomy is used which classifies the data and helps in the visualization process. The visualization used, is to represent a small ball for each information entity and to present green balls for each category. In Figure 7 an example using job offers is presented. It shows that in the category *Technology* there are the most vacancies. Overlappings references vacancies, that are applicable to more that one category. This example does not use a metaphor as it is defined, but Fluit states in [5] that such kind of visualization supports humans in creating a mental map (which we would do anyway but such drawings help).

The above given examples have one thing in common: they use information which does not carry complex semantics; only the last example uses a simple taxonomy. In all the other examples, the semantic is created externally by using similar terms or the users mind. The first two examples use visual metaphors to attract the user and to keep the user interface familiar for the most users, where as the third uses a kind of recursive digging of circles representing classifications of information.

Knowledge visualization has the additional feature that the information representing the knowledge is augmented with semantics by the information authors. This process is the subject of the next section.

[5] Spectacle is currently distributed by Aduna (http://www.aduna.com)

HOW TO ENRICH INFORMATION WITH SEMANTICS

To start this section the terms knowledge and information should be examined and a clear dividing line between them shall be drawn:

- Information is defined in [11] as "…the communication or reception of knowledge or intelligence" or "...knowledge obtained from investigation, study, or instruction...". In this article the word information refers to some data that, if received by someone, can cause an increase of knowledge. This is only possible if the context of the piece of information is known and can therefore be interpreted.
- Knowledge is defined in [11] as "...the fact or condition of knowing something with familiarity gained through experience or association" or "...acquaintance with or understanding of a science, art, or technique". The terms *understanding* or *familiarity* refer to a context to which knowledge is embedded to be useful and to be able to make inferences out of it.

But where is the difference? The difference is there where the process of knowledge visualization takes place. To gain knowledge out of information an ontology or, if sufficient the simpler taxonomy, must be applied – this is the bridge between semantics, knowledge and information. The final missing part is a way to transport knowledge. Why do not use the most powerful interface to the humans mind available – the sense of sight.

The way of generating semantics

A good approach for proper engineering of software or systems is always to look at the world or universe in which the problem resides and try to model this world in a way that the problem can be solved. It must always be considered that the machine build by humans interacts with the world on some places but it will always be a small part of the original and can never model the entire world. This was pointed out by Jackson in [21].

The above statement is applicable to semantics as well. There does not exist an ontology for the whole world or the entire universe. An ontology can only model details of the world which are of interest. The example form above with the ontology about pizzas is a good one in this context. It shows an ontology only applicable to the world of a pizzeria confronted with the problem of backing pizzas.

If starting to build an ontology, first a domain expert is needed, someone who has the ontology of the specific part of the world stored in her/his mind (e.g. how to bake pizzas). This domain specific knowledge can then be used to model the problem domain in a way that correct solutions can be inferred. Taking again the pizza example this means that to run a pizzeria someone must be aware of the pizza ontology. This can not only be used to check already created pizzas against it to see if they are correctly done, it can further be used to build up a visualization interface that fit the ontology and can possibly use a metaphor for

display. In the case of pizzas the best metaphor would be a picture of a pizza itself because most people are familiar with pizzas and they know much about the ingredients. Pizzas are good to introduce the term ontology but they are not complicated enough to think about a metaphor for them.

If the ontology is a description of a part of the world which is not as clear as pizzas to the interested audience, a good metaphor can help in convoy the knowledge derived by combining information with the applicable ontology.

FUNCTION	KNOWLEDGE TYPE	RECIPIENT	VISUALIZATION TYPE
Coordination	Know-what	Individual	Sketch
Attention	Know-how	Group	Diagram
Recall	Know-why	Organization	Image
Motivation	Know-where	Network	Map
Elaboration	Know-who		Object
New Insight			Interactive Visualization
			Story

Figure 8: Knowledge visualization framework [4]

In [4] a framework for knowledge visualization is introduced that should help a domain expert to derive a visualization out of a semantic annotation. The framework states that four perspectives must be considered in a knowledge transfer situation.

1. the *Function Type* perspective asks for the aim of a visualization and offers six possibilities shown in Figure 8.
2. the *Knowledge Type* perspective asks for the nature of knowledge to be presented because it is a difference if facts or cause and effect relationships should be visualized.
3. the *Recipient Type* perspective ask for the context in which the knowledge shall be presented. This type includes the experiences, education, cultural background and motivation of the target audience.
4. the *Visualization Type* defines a selection of available methods to visualize knowledge: *sketching* allows to visualize a basic idea quickly, they are open for interpretation; *diagrams* are abstract semantical representations that allow to demonstrate clearly defined relationships; *images* may show impressions, expressions or the real world, they stir up emotions and recall memories; *maps* represent cartographic relationships and conventions and offer an overview; *objects* are spatial and haptic, they allow to view an object form different perspectives and help to understand it (like a skeleton in biology); *interactive visualizations* allow to walk through the information or knowledge presented; *stories* create mental pictures. The phrase *reading is the cinema in your head* explains this quite well. Additionally videos could be used to tell a story about something to help in understanding.

The framework is a tool helping in the process of creating a vehicle to transfer knowledge using domain experts to generate the most appropriate visualization. The interested reader may be referred to [4] where this procedure is described extensively. The next chapter presents some examples which have been created using the introduced framework.

EXPLICIT USAGE OF SEMANTICS AND METAPHORS

Burkhard and Eppler [3,4,17,18] have explicitly used semantics and metaphors to visualize and transport knowledge. This section explains their result with some examples.

The first example is about presenting the metaphor of a tube map to help communicate complex projects to the stakeholders. This example has been published in [17]. The next example is about an iceberg chart representing risks which has been taken from [3]. In the last example, taken from [18], again a tube map is presented. There it is used as a metaphor to document the progress and information sources of a complex project.

Tube map as a metaphor for a project plan

The project has been done in an educational center for health professionals. The aim was to communicate a complex project scenario to different target groups. Since the project was about a quality certification it was essential that everyone in the organization could gain the necessary understanding to be motivated to help in getting certified.

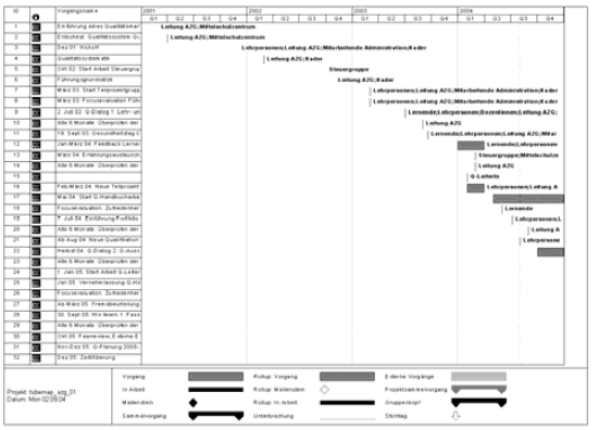


Figure 9: Gantt chart produced with MS Project®

The tube map metaphor in Figure 10 shows quite well the process of creating the most appropriate visualization out of existing knowledge. The alternative representation for the same knowledge could be a Gantt chart like it is produced by MS Project® and is shown in Figure 9.

To understand such a Gantt chart the reader would have to understand how to interpret it, s/he would need domain know how; at least how to read such diagrams (the semantics presented e.g. with an ontology). Additionally the audience is a quite large group and needs to be motivated to

understand the content of the diagram. This is definitely not the case with very heterogeneous groups of people. With the Gantt chart a domain expert would be able to create a kind of semantic, maybe in form of an ontology, to allow the automatic transformation of a MS Project® Gantt chart into a tube map metaphor.

If the previously introduced framework is used, it outputs the advice to use an easily understandable map metaphor for transferring the knowledge in the Gantt chart to the intended people.

Figure 10: Tube map visualizing a complex project [4]

To create such a metaphor several aspects have to be considered. During his studies, Burkhard pointed out in [17] five important questions that should help to cover all of them:

- Attention: How do generate attention of several people in different target groups with different educational background?
- Motivation: How to force discussion and therefore the people to partake actively in the project?
- Overview: How can an overview of the project be presented helping the people to follow the process from the beginning to the end and let them know the current status whenever they request it?
- Details: Although an overview is presented important details should not get lost.
- Strong metaphor: Since a certification process does not carry a self describing visual identity, a metaphor should be used.

The project group decided to use the metaphor of a tube map to convoy the knowledge of the entire process because of the following reasons:

- A tube system is a highly complex, dynamic and vital system.
- Such a system helps people to reach their target.
- Tube systems are publicity known and often fascinating to urban people.
- It presents an overview and details in one image.
- It structures knowledge and allows a zoom in on demand as been identified as a very important fact by Shneiderman in [8].

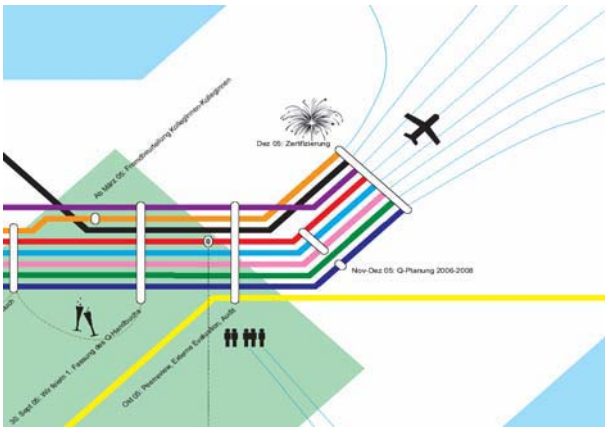


Figure 11: Zoom in with details [4]

The intention of the tube map was to reach several people that do not necessarily have a computer based job, therefore the project group decided to create a poster. A poster in this case does not only has the effect of reaching more people if mounted at places where many people meet, it further supports the metaphor of a tube map – most people know it from the subway as poster mounted in the carriage above the door. The entire poster is shown in Figure 10 whereas a detailed zoom can be seen in Figure 11.

The tube lines represent single project groups and each station represents an individual or collective milestone.

The poster was mounted at highly frequented places, like in front of the elevators as shown in Figure 12.

Figure 12: Placement of the tube map at meeting places [17]

Results:
The tested hypothesis was, if a tube map is an applicable metaphor to communicate the necessary knowledge of a complex project to different target groups. To get the result a questionnaire was used.

The result was that most people understood the metaphor and gained knowledge out of it. Further it was shown that the amount of detail is a key issue in this approach which is still not really solved. Some people complained about the mass of detail whereas others missed some milestone specific information. Another disadvantage of the poster approach was that it is very static and dynamic changes in

the project structure cannot be easily updated in the metaphor.

Iceberg chart as metaphor for risk management

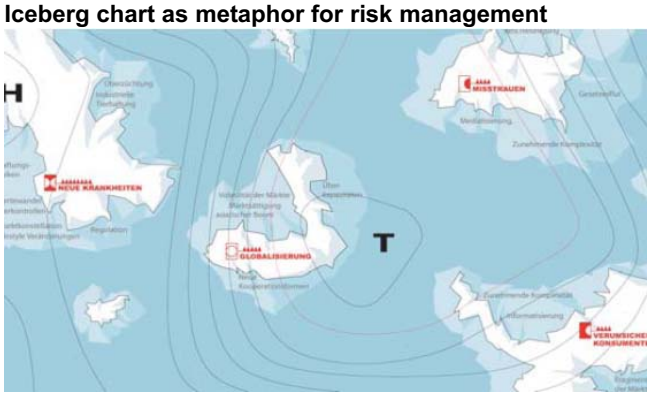


Figure 13: Iceberg chart representing risks [3]

The in [3] presented visualization of an iceberg chart uses the metaphor of icebergs swimming in the sea representing risks[6] (everyone knows the film "Titanic"). The size of each iceberg represents its risk whereas not only the visible part of an iceberg should be considered. Under the surface the risk drivers are shown. In other words the drivers under the surface build up a risk which keeps the risk above the water. A picture is shown in Figure 13.

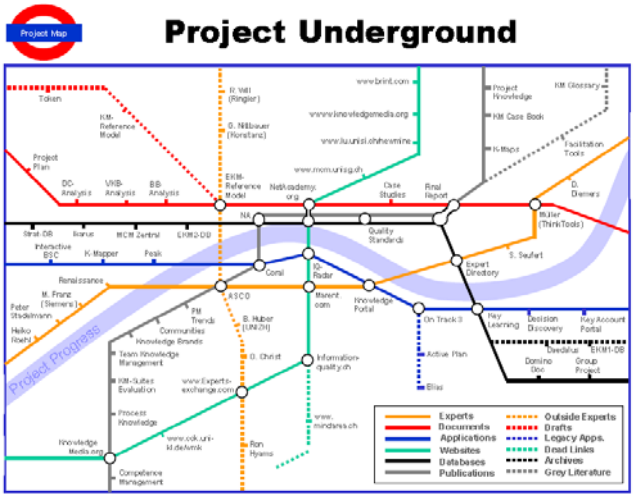


Figure 14: Metro Map representing a finished project [18]

Another tube map as metaphor for a mind map

An example taken from Eppler [18] is a project underground/tube map. The project underground is a metaphor for a mind map showing information sources needed in a project environment. In this drawing (see Figure 14) each metro line indicates an information source which could be a document, a human expert, a database, software applications, websites or scientific articles. The river running through the city indicates the project progress. This interactive map was used to document an already finished project and links the various results of the project visually to each other.

Examples found on the Internet

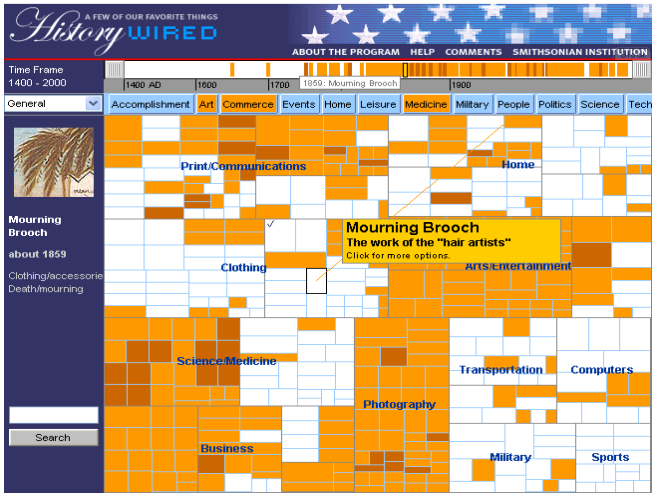


Figure 15: History wired, a museum site introduces visitors to more than three million objects (http://historywired.si.edu/)

As knowledge visualization and the access to large digital libraries go hand in hand using metaphors, a few examples can already be found on the web if looking carefully. One interesting example is the search interface of the National Museum of American History, Behring Center in Washington, D.C. There the metaphor of a city map of an North American city is used to explore the more than three million objects held there. Basis for this example is a kind of tree map introduced by Johnson and Sheideman in [22]. It is as well an example that visualization methods usually used to visualize relationships in large digital libraries, could adapted to help end users to navigate through a huge amount of data. As semantics, as usual for a museum, taxonomies are used whereas an additional time dimension as indicated by the timeline is used to help people to find timely related objects in the museum.

WHO WILL GAIN BENEFIT FROM KNOWLEDGE VISUALIZATIONS USING METAPHORS?

The concept of transferring knowledge to users is the aim of almost all interactive applications. In many cases applications are used in a given context, e.g. applications in an organization for special purposes, like documentation, or accounting. These applications are mostly operated by especially trained personnel and the effect of metaphors would make their work more cumbersome than easier. Metaphors are more applicable to applications or parts of applications that are used by people not familiar with what is shown there. This could be the reporting interface of an accounting application or applications on the web which must be attractive to the user and give the visitor a feeling of knowing what is all about. This feeling makes it more

[6] The visualization has been created by vasp datatecture GmbH, Zürich, Switzerland.

likely that a visitor comes back. As pointed out in [18] knowledge visualization in organizations can help in strategic management, advertising and marketing. Today most of the available budget for visualizations is spend for advertisements and cooperate identity. In advertisements, it is common to address emotions of the audience by telling a story or using metaphors to communicate sometimes complex messages in a very simple way. But both advertisements and marketing does not explore the entire potential of knowledge visualizations. The techniques introduced in this article can be used in any case where knowledge is to be transferred. This could be complex strategic decisions that are based on knowledge gained form visualizations or knowledge needed for understanding complex relations in engineering or even to communicate simple knowledge to many people (if the fact to inform a majority is important).

KNOWLEDGE VISUALIZATIONS USING METAPHORS IN INTERACTIVE SYSTEMS

Already in 2001 Chao introduces in [20] a novel interface for computer administration using a metaphor – he introduced the first-person shooter game *doom* as a 3D interface for that task. The metaphor of such a game is used due to the similarity of killing processes on a computer. Although the project itself has never been used in an application, the interest of the community was enormous. The project website counted about several thousand visitors after the first weeks of being online, reflecting the interest on such interfaces.

According to Burkhard [4] even knowledge visualizations will go off the screen. In the beginning of such visualizations they were often paper based and used as printouts to attract the audience. More and more these techniques get use of more advanced user interfaces. *Moving from "painted bits" on rectangular screens to "tangible bits" in a 3D environment* or *using the opulence of human senses and skill for a more comprehensive usage of knowledge transfer technologies* [4] are some statements in this context. In the first sections the difference of information and knowledge has been introduced by a simple example: information was referred as a set of facts like a telephone number whereas knowledge has been introduces at something that must be learned and augmented with a specific domain know how. The example given was the task of learning to ride a bicycle. With tangible user interfaces even such knowledge could be transferred more easily and flexible.

SUMMARY

Knowledge visualizations technologies and the research on user interfaces for large digital libraries go hand in hand in many cases. This is because both research fields have learned that technologies offered by the semantic web are helpful to build a bridge between computers and humans, which use completely different ways of storing and retrieving information. Computers must be told how to infer additional knowledge out of already available information and knowledge, a process that is accomplished by humans during their education and everyday live. Both fields, digital libraries and knowledge visualizations, are currently two hot topics in research.

However a clear line must be drawn between knowledge visualization and information visualization. Although the presentation of information and knowledge uses the same approaches (metaphors, 3D interfaces, semantics, etc.), both have another aim. Information visualization aims to explore abstract data available in huge digital libraries and to help people to navigate through this enormous amount of information. Examples given for that are search engines like KartOO, Ujik or Grooker respectively the interface to the museums content of History Wired (which is somehow in between because it provides information about objects and knowledge about epochs where they have been used).

Knowledge visualization in contrast has the aim to transfer and improve knowledge among people. It gives people the ability to express knowledge in a way that others could more easily learn or gain new insights of already understood. Examples for that are given with the visualization of a tube map representing the progress of a complex project that must be beard by many different people within an organization or the iceberg chart which indicates strategic risks that an organization is facing.

Both research fields build up on semantics explained in the first sections of this article. Important terms like ontologies or taxonomies have been introduced with understandable examples like the first taxonomy in biological science by Carl von Linné or the pizza ontology. Further a way of thinking how semantics can be created and a short introduction and categorization of metaphors is given.

To close this article it can be pointed out that knowledge visualization is a very interesting new research field offering researchers to deal with visual representation (this does not exclude tangible representations) and research in field of semantics with its interesting excurses in human psychology of learning, the cognitive psychology or the neuroscience.

REFERENCES

1. Lamping, J. and Rao, R. (1994): "Laying Out and Visualizing Large Trees Using a Hyperbolic Space." In: Proceedings of the ACM Symposium on User Interface Software and Technology, ACM Press, 1994, Page 13-14.
2. Berners-Lee, T., et.al. (2001): "Semantic Web Services." In: Scientific American Magazine, May 2001.
3. Burkhard, R. (2006): „Knowledge Visualization: Die nächste Herausforderung für Semantic Web Forschende." In: Semantic Web, Wege zur vernetzten Wissensgesellschaft, Springer-Verlag Berlin Heidelberg, 2002, Page 201-211.
4. Burkhard, R. (2005): "Knowledge Visualization - The Use of Complementary Visual Representations for the Transfer of Knowledge. A Model, a Framework, and Four New Approaches." Dissertation ETH Zürich, 2005
5. Fluit et. al. (2003): "Ontology-based Information Visualization." In: Visualizing the Semantic Web: XML-based Internet and Information Visualization, Springer-Verlag London, 2006, Page 36-48.
6. Börner K. and Chen, C. (2002): "Visual Interfaces to Digital Libraries." In: LNCS 2539, Springer-Verlag Berlin Heidelberg, 2002, pages 1-9.
7. Bush, V. (1945): "As we may think." In: interactions, volume 3, number 2, ACM Press 1996, page 35-46
8. Shneiderman, B. (1996): "The eyes have it: a task by data type taxonomy for information visualizations." In: Proceedings of the IEEE Symposium on Visual Languages, 1996 pages 336-343
9. Daconta, M., et. al. (2003): "The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management." John Wiley & Sons, Indianapolis (2003)
10. Gruber, T. R. (1995): "Toward principles for the design of ontologies used for knowledge sharing." In: International Journal of Human-Computer Studies, Vol. 43, Issues 4-5, November 1995, pages 907-928.
11. Merriam-Webster online edition at http://www.m-w.com/, visited in April 2007
12. Horridge, M., et. al. (2004): "A Practical Guide to Building OWL Ontologies Using the Protègè-OWL Plug-in and CO-ODE Tools." The University Of Manchester, 2004
13. Schreiber, P. (2005): "Was ist Bildwissenschaft? Versuch einer Standort- und Inhaltsbestimmung." In: Image: Zeitschrift für interdisziplinäre Bildwissenschaft, Herbert von Halem Verlag, Köln, 1/2005, pages 5-9
14. Horn, R. (1999): "Information Design: Emergence of a New Profession." In: Information Design by Jacobson (Editor), MIT Press, 2000, pages 15-34
15. Baleydier, L. and Baleydier, N. KartOO search engine at http://kartoo.com/, visited in Mai 2007
16. Baleydier, L. and Baleydier, N. Ujiko search engine at http://ujiko.com/, visited in Mai 2007
17. Burkhard, R., et. al. (2004): "Tube Map: Evaluation of a Visual Metaphor for Interfunctional Communication of Complex Projects." In: Proceedings of I-KNOW '04, Graz, Austria, June 2004, pages 449-456.
18. Eppler, M.J., et. al. (2004): "Knowledge Visualization - Towards a New Discipline and its Fields of Application." In: Lecture Notes in Computer Science, Volume 3426/2005, Springer Verlag Berlin, 2005, pages 238-255.
19. Grooker enterprise search management (2007) at http://www.groxis.com/, last visited in May 2007
20. Chao, D. (2001): "Doom as an Interface for Process Management." In: Proceedings of the CHI2001 Conference on Human Factors in Computing Systems 2001, Seattle, pages: 152 – 157.
21. Jackson, M, (1995): "The world and the machine." In: Proceedings of the 17th international conference on Software engineering, 1995, Seattle, pages: 283 – 292.
22. Johnson, B., Shneiderman, B. (1991): "Tree-Maps: a space-filling approach to the visualization of hierarchical information structures." In: Proceedings of the 2nd conference on Visualization, 1991, San Diego, pages 284 – 291.
23. OWL Web Ontology Language Reference. W3C Recommendation 10 February 2004. URL: http://www.w3.org/TR/2004/REC-owl-ref-20040210/, last visited in June 2007.

Softwaredesign von Werkzeugen für interaktive fortgeschrittene (2D) Informationsvisualisierungen

Nadja Marko
Institut für Informatik-Systeme
Universität Klagenfurt
nmarko@edu.uni-klu.ac.at

ABSTRACT
In den letzten Jahren haben sich immer mehr Daten und Informationen angesammelt. Mit den herkömmlichen Methoden ist es sehr schwierig diese große Menge an Information übersichtlich darzustellen. Um das Problem zu lösen, wurde eine Reihe von fortgeschrittenen Visualisierungsmethoden entwickelt, die die Darstellung von strukturierten und unstrukturierten Daten übersichtlicher machen sollen. Doch trotz der intensiven Forschung und Weiterentwicklung an diesen fortgeschrittenen Visualisierungen haben sie sich bei Entwickler von Anwendungen und den Benutzern nicht wirklich durchgesetzt. In dieser Arbeit sollen mehrere dieser Visualisierungsmethoden kurz aufgezeigt werden. Anhand verschiedener Merkmale wird besprochen warum sie nicht standardmäßig verwendet werden, wie zum Beispiel einfache Bäume und Tabellen. Dabei soll das Hauptaugenmerk auf der Untersuchung der Architektur von Tools liegen. Die Schwierigkeiten und die hohen Anforderungen an das Softwaredesign sollen aufgezeigt werden. Der letzte Teil befasst sich mit der Architektur zweier Tools, die die Programmierung von einigen fortgeschrittenen Visualisierungsmethoden unterstützen. Die Funktionen, die Programmiermöglichkeiten und das Design der Tools werden betrachtet.

Author Keywords
Fortgeschrittene Visualisierungsmethoden, Architektur, Visualisierungstool, prefuse, InfoVis.

ACM Classification Keywords
HCI, Software design. Infomationsvisualisierungen

EINLEITUNG

Die Menge an Informationen, die wir regelmäßig anwenden müssen, wächst immer mehr. Es ist sehr schwierig geworden diese Informationsfülle effektiv zu verarbeiten. Wenn man nur an das Internet denkt, erweist es sich sehr schwierig die „richtigen" Informationen herauszufiltern. Methoden zur Darstellung von Abhängigkeiten zwischen Daten gibt es schon länger. Bäume, Graphen und Tabellen sind mittlerweile für Jedermann ein Begriff. Doch für die Wiedergabe der immer größer werdenden Datenmenge reichen diese herkömmlichen Methoden nicht mehr aus. Einfache Bäume und Graphen spiegeln nur einen Ausschnitt der Daten wider. Die Übersicht geht dabei verloren. Seit einigen Jahren forscht man in diesem Bereich und strebt nach der Entwicklung einer effektiven Methode, um eine große Datenmenge überschaubar darzustellen. Nebenbei soll es weiters möglich sein Details von relevanten Datenobjekten zu erkennen.

Es existieren mittlerweile einige neuere Visualisierungsmethoden, die eine große Informationsmenge übersichtlich darstellen und gleichzeitig die Fokussierung auf bestimmte Datenobjekte erlauben. Laufend werden neue Techniken entwickelt, doch sie werden anscheinend nur in wenigen Bereichen eingesetzt. Es existieren bereits einige Anwendungen, die fortgeschrittene Visualisierungen von Daten ermöglichen. Diese erlauben die Darstellung von Daten nach einer bestimmten Methode. Bei Anwendern und Entwicklern scheint sich allerdings noch keines durchgesetzt zu haben. Fortgeschrittene Visualisierungsmethoden werden bis jetzt noch nicht alltäglich eingesetzt, wie zum Beispiel Buttons und Tabellen. Gründe dafür sind unter anderem die inkompatiblen Datenformate, eine inkonsistente Terminologie und die nicht standardisierten Widgets [6].

Damit sich neue Informationsvisualisierungen etablieren, können unterstützende Tools förderlich sein. In den letzten Jahren wurden einige Werkzeuge gebaut, die die Entwicklung von mehreren neuen Visualisierungsmethoden ermöglichen. In dieser Arbeit wird ein Vorschlag für das Softwaredesign von Werkzeugen, die die Implementierung von Visualisierungen erleichtern sollen, erklärt. Dieses Referenzmodell für das Design wird von Shneiderman et Al. in [6] vorgestellt. Außerdem sollen die Schwierigkeiten, die sich bei der Entwicklung von solchen Anwendungen ergeben, erörtert werden.

Das *InfoVis* [14] und das *prefuse* [7] Toolkit sind zwei Werkzeuge, die Informationsvisualisierungen nach verschiedenen Methoden ermöglichen. Beide sind in Java Swing Anwendungen integrierbar und können die Programmgröße, die für die Implementierung von visuellen Darstellungen gebraucht werden, erheblich reduzieren. Das Design und die Funktionsweise der beiden Tools sollen in dieser Arbeit betrachtet und abschließend nach bestimmten Kriterien beurteilt werden.

Die Arbeit gliedert sich in folgende Teile: Im nächsten Kapitel werden einige der bekanntesten fortgeschrittenen Visualisierungsmethoden kurz erklärt. Der dritte Abschnitt beschäftigt sich mit dem Design von Visualisierungswerkzeugen, den Herausforderungen, die sich damit ergeben und einigen Kriterien, die zur Beurteilung von Werkzeugen herangezogen werden können. Im vierten Teil werden die beiden Toolkits *InfoVis* und *prefuse* erläutert. Das letzte Kapitel liefert eine Zusammenfassung, in der die wesentlichen Schwierigkeiten der Anwendungskonstruktion noch Mal verdeutlicht werden.

FORTGESCHRITTENE VISUALISIERUNGSMETHODEN

Für die Darstellung einer großen Informationsmenge existieren mehrere Möglichkeiten. Man unterscheidet prinzipiell zwischen 1D, 2D und 3D Darstellungen und Bäumen und Graphen. Mit fortgeschrittenen Visualisierungsmethoden versucht man viel Information so übersichtlich und gleichzeitig so detailliert wie möglich darzustellen. In diesem Abschnitt soll das Konzept einiger neueren Visualisierungsmethoden kurz erklärt werden. Zu den Visualisierungen, die die Raumausnützung verbessern sollen, gehören: *RadialLayout*, *TreeMaps* [15], *HyperbolicTrees* [12], *Degree-of-Interest Trees* (DOITrees) [2], *SpaceTrees* [10] und *StarfieldDisplays* [16]. Außerdem werden die zwei *Fokus + Kontext* Techniken *FisheyeViews* [17] und *BifocalLens* [18] betrachtet, die gleichzeitig detaillierte Informationen und die Übersicht über die Daten gestatten.

Räumliche Visualisierungen

RadialLayout. Ein Baum hat ein radiales Layout, wenn die Kanten nicht nur in eine Richtung zeigen, sondern sich rund um die Wurzel formieren. In Abbildung 1 sieht man einen Baum, der im RadialLayout gezeigt wird.

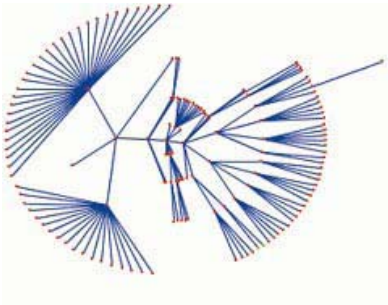


Abbildung 1: RadialLayout von einem Baum [9]

TreeMaps. TreeMap ist eine Visualisierung, die sehr viel Platz am Display nützt. Baumknoten werden mittels Vierecken angezeigt. Die Größe der Vierecke spiegelt die Größe des Knoten wider. Die Unterordnung von Baumknoten stellt man mit der Unterteilung des Knotens dar. Ein Viereck beinhaltet somit weitere Vierecke (Söhne). Labels und Farben werden für die Lesbarkeit des Baumes benutzt. Abbildung 2 zeigt die Idee von TreeMaps und Abbildung 3 veranschaulicht, wie ein Baum als TreeMap aussieht. Bei einem Benutzerexperiment, das von Kobsa in [1] diskutiert wird, haben TreeMaps bezüglich Interpretation und Benutzerzufriedenheit gut abgeschnitten. Aufgaben, die den Benutzern gestellt wurden, konnten mit TreeMaps korrekt und in angemessener Zeit ausgeführt werden.

Abbildung 2: TreeMap [9]

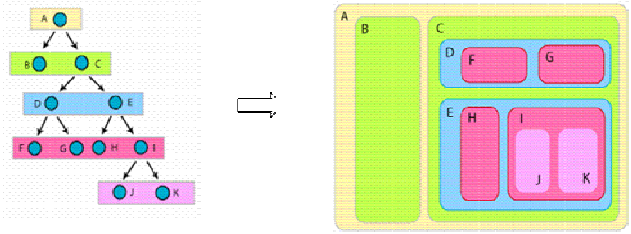


Abbildung 3: Baum als TreeMap [8]

DOITree. DOITrees werden wie die Bäume dargestellt, die wir schon kennen. Doch zusätzlich unterscheidet man eine kleine Anzahl an Knotengrößen, zum Beispiel: klein, mittel und groß, für die Kennzeichnung der Fokusknoten. Je nach Größe des Knotens werden unterschiedliche Informationsmengen angezeigt. Außerdem ist es möglich, durch ungleiche Darstellungen der Knoten (zum Beispiel die Farbe betreffend), zusätzliche Informationen zu speichern [2]. Der Knoten auf dem der Fokus liegt und die Vorgänger dieses Knotens bis hin zur Wurzel werden größer dargestellt. Die Größe der verbleibenden Knoten sinkt linear mit der Distanz zum Fokus. Abbildung 4 zeigt einen DOITree mit Fokus auf Knoten 3. Hier kennzeichnet man den Fokusknoten zusätzlich durch eine andere Farbe (blau). Abbildung 5 zeigt den gleichen DOITree wie in Abbildung 4 mit erweitertem Fokus auf einem Blattknoten.

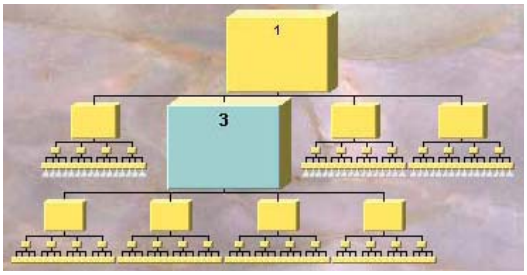


Abbildung 4: DOITree mit Fokus auf Knoten 3 [2]

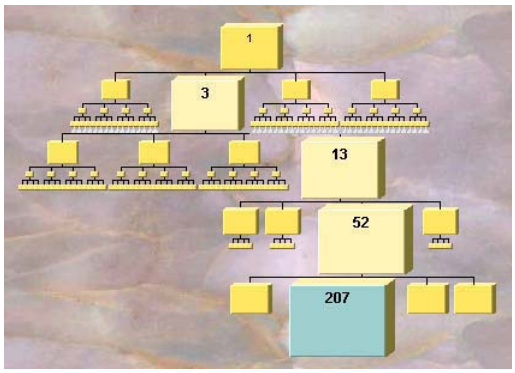


Abbildung 5: Der gleiche DOITree mit Fokus auf einem Blattknoten [2]

HyperbolicTree. Die Idee des HyperbolicTrees ist, dass die Anzeigefläche eines Knoten kontinuierlich mit der Distanz zum Fokus sinkt. Es werden aber noch so viel Eltern und Kinderknoten angezeigt, dass man die Übersicht nicht verliert. Zu Beginn liegt der Fokus auf der Wurzel des Baumes. Durch Manipulationen kann man den Fokus dann ändern [12]. In Abbildung 6 sieht man das Konzept eines HyperbolicTrees. Zum Rand hin sinkt der Platz für die Knoten während in der Mitte (Fokus) die Anzeigefläche am Größten ist.

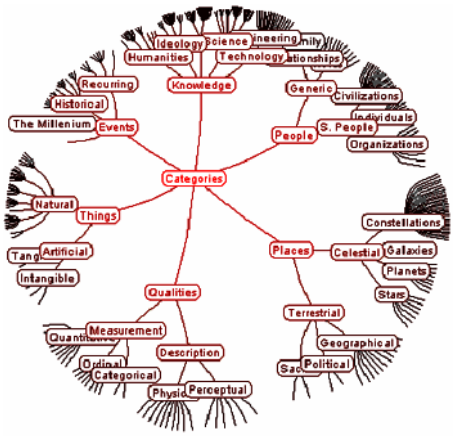


Abbildung 6: HyperbolicTree [13]

SpaceTree. Mit SpaceTrees (Abbildung 7) werden große Bäume folgendermaßen angezeigt: Zweige, die nicht auf die Anzeigefläche passen, fasst man mit einem Dreieck zusammen. Die Schattierung des Dreiecks ist proportional zur gesamten Anzahl der Knoten im Subbaum. Dunklere Dreiecke entsprechen Zweigen mit mehr Knoten während hellere Dreiecke Zweige mit weniger Knoten gleichkommen. Die Höhe des Dreiecks präsentiert die Tiefe und die Basis die durchschnittliche Breite des Subbaumes. Außerdem kann man auswählen, ob die Vorschau relativ zur Wurzel oder relativ zum Vaterknoten angezeigt werden soll [10].

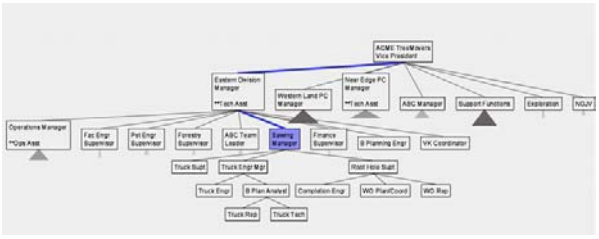


Abbildung 7: SpaceTree [11]

StarfieldDisplay. Ein StarfieldDisplay ist eine weiterentwickelte Form der Streudiagramme mit zusätzlichen Funktionen für die Auswahl und das Zoomen von Daten [16]. Diese Visualisierungsmethode wurde für die Darstellung von Datenbankinhalten entwickelt. Die Punkte in diesem Diagramm repräsentieren einen Datensatz in einer Datenbank. Die x-Achse und die y-Achse präsentieren dabei ein Ordnungskriterium, anhand dessen die Datensätze im Raum eingegliedert werden können. Zum Beispiel könnten die Achsen in einer Personendatenbank *Alter* und *Gehalt* sein. Personen höheren Alters und mit höherem Einkommen befinden sich im Streudiagramm dann im rechten oberen Teil. Man kann zusätzlich unterschiedliche Formen und Farben verwenden und somit die Komplexität der Daten darstellen. Abbildung 8 zeigt ein mögliches StarfieldDisplay.

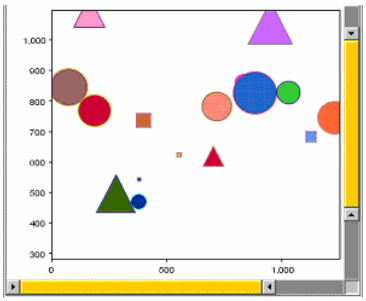


Abbildung 8: StarfieldDisplay [13]

Fokus + Kontext

Bei großen Datenmengen besteht das Problem, dass nicht alle Daten auf die Anzeigefläche passen. Bei herkömmlichen Methoden, wie scrollen, verliert man schnell den Überblick. *FisheyeViews* und *BifocalLenses* sind zwei fortgeschrittene Techniken, die es ermöglichen genauere Informationen anzuzeigen, ohne dass man die Übersicht verliert. *Fokus* bedeutet in diesem Zusammenhang eine detaillierte Betrachtung von Objekten und *Kontext* die übersichtliche Darstellung.

FisheyeViews. Die Idee bei FisheyeViews ist, dass Objekte, die sich im Zentrum des Displays befinden, am größten angezeigt werden. Objekte die sich am Rand aufhalten, werden hingegen immer kleiner. In der Mitte des Bildes sind wenige Objekte, die genauer betrachtet werden. Zum Rand hin gibt es viele, aber ohne Details. Dabei verzerrt (biegt) man die Sicht um den Mittelpunkt. In Abbildung 9 sieht man zum Beispiel ein Fisheye Menü.

Abbildung 9: Fisheye Menü [7]

BifocalLens. Die Idee dieser Methode kommt von der Optik. Bei der Visualisierung über bifokale Linsen verwendet man zwei Levels von Displays. Einen Level nutzt man für die Übersicht und einen für detaillierte Informationen. Die detaillierte Sicht auf die Objekte wird in die Übersicht (in die Mitte) eingefügt. Abbildung 10 veranschaulicht die Idee von bifokalen Linsen. Die Anzeigefläche wird rechts und links vom Fokus gebogen. Den Fokus stellt man dann in der Mitte detailliert dar. Der Kontext befindet sich seitlich vom Fokus in kleinerer und dafür übersichtlicherer Form.

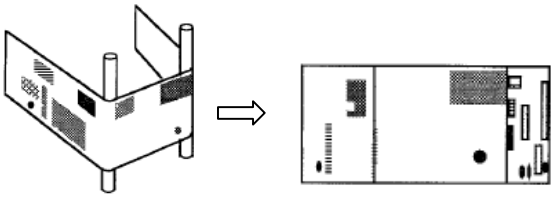


Abbildung 10: Idee bifokale Linse [18]

Methoden, die diese Visualisierungen ermöglichen, sind: Filtern, selektive Aggregation, Micro-Macro-Readings, Highlighting und Verzerrung [6].

Die Entwicklung neuer Visualisierungen ist nicht leicht. Im nächsten Teil werden einige Schwierigkeiten davon erwähnt.

Schwierigkeiten

Beim Entwickeln von neuen Informationsvisualisierungen bleiben einige Diskussionspunkte und Schwierigkeiten, die noch bewältigt werden müssen. Nach Shneiderman et. Al. in [6] sind solche Punkte, die noch geklärt werden müssen:

2D versus 3D Präsentationen. Ein gängiges Diskussionsthema ist der Einsatz von 2D oder 3D Informationsvisualisierungen. Beide haben Vor- und Nachteile und eignen sich für verschiedene Visualisierungen unterschiedlich gut. Zum Beispiel werden von den Benutzern 3D Darstellungen für Geschäftsgrafiken bevorzugt und Bäume werden in 2D als nützlicher empfunden.

Übersicht* + *Detail* versus *Fokus* + *Kontext. Ein weiterer umstrittener Bereich ist die Strategie für das Behandeln von großen Informationsräumen. Eine Anwendungsmöglichkeit ist das Zoomen, das für Benutzer leicht verständlich ist, aber bald unübersichtlich wird. Mit der *Übersicht + Detail* Strategie hat man ein Fenster für die Übersicht und ein zweites für die detaillierte Betrachtung [6]. *Fokus + Kontext* zeigt Details in einem Fokuspunkt, der in der Übersicht enthalten ist, wie im vorherigen Abschnitt bereits besprochen. Dabei sind Zoomfaktor und Skalierbarkeit der Strategien ein wesentliches Maß.

Überarbeiten des Desktops. Für fortgeschrittene Informationsvisualisierungen wird der Desktop drastisch verändert, was eine schwierige Aufgabe darstellt. Auch bei diesem Punkt gibt es verschiedene Techniken, die mehr oder weniger gut für verschiedene Visualisierungen geeignet sind.

Bewältigen der Mehrfach-Werte Attribute. Eine Herausforderung betrifft Datenmengen, die Attribute mit Mehrfach-Werten enthalten. Das Darstellen von solchen Attributen erweist sich als mühsame Aufgabe.

Verstehen der menschlichen Auffassung. Hier stellt man sich Fragen wie: Wie benutzt man Größe und Farbe für die Interpretation? Wann sind Animationen sinnvoll? Wie kann Zoomen leicht und für den Benutzer verständlich machen?

Es gibt verschiedene Kriterien, die man für die Evaluierung des Layouts der Visualisierungen untersuchen kann. In [2] sind die drei Hauptkriterien: Einfachheit der Interpretation, räumliche Organisation und Benutzerpräferenzen. Die Kriterien zu erfüllen, stellt bei der umfangreichen Datenmenge oft eine große Aufgabe dar.

Damit sich fortgeschrittene Visualisierungen durchsetzen, ist es hilfreich, dass Werkzeuge angeboten werden, die die Entwicklung unterstützen. Im nächsten Kapitel werden die Herausforderungen, die sich damit ergeben, dargelegt. Weiters wird ein Vorschlag für das Softwaredesign beschrieben, das die hohen Anforderungen an solche Werkzeuge erfüllen soll.

ENTWICKLUNG VON ANWENDUNGEN

Die Entwicklung von Tools, die (neuere) Visualisierungsmethoden unterstützen, ist eine schwierige Aufgabe. Wenn SpaceTrees, TreeMaps und dergleichen eingesetzt werden sollen, müssen Anwendungen existieren, die ein schnelles Feedback liefern und leicht zu bedienen sind.

Herausforderungen

Die Herausforderungen, die Anwendungen zur Informationsvisualisierung bewältigen müssen, sind [5]:

Importieren der Daten. Damit ist das Importieren der Daten von Datenbanken oder Dokumenten für die Informationsvisualisierung mit dem Tool gemeint. Die Entscheidung, wie man Daten innerhalb der Anwendung organisieren soll, um gute Visualisierungsergebnisse zu erzielen, ist eine schwierige Aufgabe. Die Daten in das für das Tool richtige Format zu bringen, unzulässige Datensätze herauszufiltern und fehlende Daten zu erstellen sind langwierige Aufgaben mit denen man sich befassen muss.

Kombinieren von visuellen Darstellungen mit Labels. Die Darstellungsart der Informationen hat sehr viel Aussagekraft, aber textuelle Beschreibungen der Daten dürfen nicht vernachlässigt werden. Labels sollten gut lesbar sein und die Anzeigfläche trotzdem nicht überschwemmen.

Wahrnehmen von zusammengehörenden Informationen. Es sollte möglich sein Informationen, die miteinander in Beziehung stehen, gut und schnell zu erkennen. Man

braucht dafür einen schnellen Zugang zu (einer großen Menge) zusammengehörender Informationen.

Betrachten von großen Datenmengen. Das Verarbeiten der riesigen Datenmengen von bis zu einigen Millionen von Attributen, ist ein generelles Problem von Visualisierungsmethoden. Die Wahrnehmungsfähigkeiten von Benutzern sind begrenzt. Eine für den Benutzer leicht interpretierbare Darstellung von der großen Informationsmenge erweist sich als äußerst problematisch (siehe vorheriges Kapitel).

Integrieren von Data-Mining. Informationsvisualisierungen und Data-Mining sind zwei unterschiedliche Forschungsgebiete. Informationsvisualisierungsexperten beschäftigen sich mit dem Wahrnehmungssystem der Benutzer und Data-Mining Experten machen sich statistische Algorithmen und maschinelles Lernen zu Nutze. Die Unterstützung der Visualisierungsmethoden mit Data-Mining Methoden, kann oft sehr hilfreich sein, um zum Beispiel Tendenzen zu erkennen.

Zusammenarbeiten mit Anderen. Die Werkzeuge sollten die Weitergabe von Visualisierungszuständen unterstützen, um sie auch anderen Personen zugänglich zu machen.

Erreichen universaler Usability. Die Werkzeuge sollten für alle möglichen Anwendergruppen leicht zu bedienen sein.

Zusätzlich zu diesen Herausforderungen beruht der Erfolg von interaktiven Benutzeroberflächen auch auf der direkten Manipulation von Daten. Es sollte bei der großen Informationsmenge unter anderem auch möglich sein Attribute zu zoomen, zu filtern, miteinander in Beziehung zu setzen und History-Aktionen durchzuführen. Das ist eine schwierige Aufgabe, da die Implementierung von komplexen Layoutalgorithmen und dynamischen Grafiken hohe mathematische Fertigkeiten und mächtige Programmierfähigkeiten erfordert [4].

Referenzmodell für die Visualisierung

Die Architektur bildet das Grundgerüst für die Anwendungen. Die Entwicklung von Visualisierungsanwendungen muss viele Herausforderungen, wie zuvor beschrieben, meistern. Um die speziellen Anforderungen für die interaktive Informationsvisualisierungen zu erfüllen, muss die Architektur flexibel aufgebaut sein. In [6] wird ein Referenzmodell für die Architektur von Visualisierungsanwendungen gezeigt, das nun verdeutlicht werden soll.

Der Aufbau des Modells wird in Abbildung 11 gezeigt. Man beginnt links mit den Rohdaten und kommt dann mit mehreren Transformationen der Daten nach rechts zu den Views. Die Pfeile, die vom Benutzer weg zu den Transformationen führen, kennzeichnen die Anpassung der Transformationen durch Benutzerinteraktion.

Abbildung Daten auf Views

Mit *Daten-Transformationen* bringt man Daten in eine strukturierte Form, was die spätere Visualisierung vereinfacht. Man bildet die *Rohdaten* auf *Datentabellen* ab. Mit

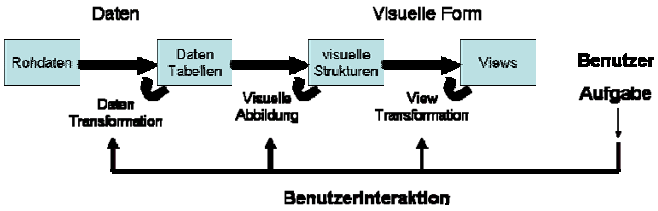


Abbildung 11: Referenzmodell für Visualisierungen [6]

Rohdaten meint man die spezifischen Formate, wie einfache Texte oder Daten von Kalkulationsprogrammen. Die *Datentabellen* sind Relationen, die aus Daten (auch Variablen sind möglich) und Metadaten bestehen. Die *visuelle Abbildung* formt die Datentabellen in *visuelle Strukturen* um. Es gibt mehrere Möglichkeiten für die Abbildung. Wenn alle Daten der Tabelle auch in der *visuellen Struktur* vorhanden sind, ist die Abbildung gut und ausdrucksstark. Die *visuellen Strukturen* kombinieren Raum, Kennzeichen und grafische Eigenschaften. Zum Schluss bilden *View-Transformationen* die visuellen Strukturen auf *Views* ab. Dabei werden Parameter wie Position, Skalierung und Ausschnitt (des Displays) übergeben, um die *Views* aus den *visuellen Strukturen* abzuleiten. Zu den *View-Transformationen* zählen zum Beispiel *Fokus + Kontext* und *Übersicht + Detail* Methoden. Durch die Interaktion mit dem Benutzer können Parameter dieser Transformationen geändert werden. Das ermöglicht zum Beispiel die Sicht auf bestimmte Daten.

Die Hauptaufgabe stellt die Transformation der, auf mathematischen Relationen basierenden Tabellen, auf die visuellen Strukturen, die grafische Eigenschaften haben, dar [6].

Interaktion

Dem Benutzer ist es möglich mit Hilfe von Interaktionskontrollen die Visualisierungen zu manipulieren. Die Interaktionskontrollen können in die Visualisierung integriert oder separat sein.

Für die Interaktion mit den Visualisierungen gibt es verschiedene Techniken. Solche, die die *Daten-Transformation* modifizieren sind *dynamic queries, direct walk, details-on-demand, attribute walk, brushing* und *direkte Manipulation* [6]. *Dynamic Queries* sind die visuelle Alternative für SQL-Abfragen. Damit ermöglicht man, zum Beispiel mit Buttons oder Sliders, die Veränderung von Variablen der Datentabelle. *Direct Walk* benutzt Verlinkungen, wie sie im Web verwendet werden. Mit mehreren Klicks auf die Visualisierung kann der Benutzer nach Informationen suchen. *Details-on-demand* ermöglicht bei der Abbildung auf die Visualisierung die Beteiligung mehrerer Variablen. Beim *attribute walk* kann der Benutzer Datensätze auswählen und nach weiteren Datensätzen suchen, die ähnliche Attribute haben. *Brushing* verwendet man bei mehreren Visualisierungen des gleichen Objektes. Wenn man einen Datensatz eines Views hervorhebt wird dieser Datensatz auch in anderen Views ausgewählt. *Direkte Ma-*

nipulation nützt man für das Modifizieren von Transformationen. Man setzt zum Beispiel Widgets dafür ein.

Interaktionstechniken, die die *visuellen Abbildungen* modifizieren sind *dataflow* und *Pivottabellen*. Bei *dataflow* verwendet man eine explizite Präsentation für die Darstellung der Abbildung. *Pivottabellen* ermöglichen eine rasche Manipulation der Abbildung auf visuelle Strukturen (vergleichbar mit Pivottabellen in Kalkulationsprogrammen) [6].

Direct selection, camera movement, magic lens, Übersicht + Detail und *zoomen* sind Methoden, die die *View-Transformation* modifizieren [6]. *Direct selection* betrifft die Auswahl und das Hervorheben von Objekten und Objektgruppen. *Camera movement* ermöglicht die Positionsänderung des Beobachters, zum Beispiel in Informations-Landschaftsräumen. Ein anderer Gesichtspunkt auf die gleichen Daten kann für die Benutzer oft hilfreich sein. Mit *magic lenses* wählt man Objekte nach der x-, y-Position und wendet daraufhin weitere Selektionsmethoden, wie *dynamic queries*, an. *Übersicht + Detail* wurde bereits erwähnt. Beim *Zoomen* werden die sichtbaren Objekte reduziert und dafür wird die Anzahl der Variablen der gezoomten Objekte erhöht.

Das Referenzmodell von Shneiderman et Al. [6] zeigt eine mögliche Lösung für das Softwaredesign von Werkzeugen zur Unterstützung von (fortgeschrittenen) Informationsvisualisierungen. Werkzeuge kann man für unterschiedliche Aufgaben nützen. Die angebotenen Funktionen unterscheiden sich von Werkzeug zu Werkzeug. Im nächsten Abschnitt werden Punkte aufgelistet, nach denen man solche Tools beurteilen kann.

Beurteilung

Software Tools für die Programmierung von Interfaces kann man nach sechs Kriterien bewerten [5]:

1. **Teile der Anwendung.** Man untersucht die Tools nach dem Unterstützungsgrad für die Entwicklung eines Interfaces. Hier betrachtet man zum Beispiel, ob nur die Entwicklung der Präsentationskomponente oder auch von generellen Programmiermechanismen unterstützt wird.
2. **Lernzeit.** Das ist die Zeit, die man für das Lernen der Toolbedienung braucht.
3. **Entwicklungszeit.** Entwicklungszeit ist die Zeit, die man für das Erstellen einer Benutzeroberfläche braucht, wenn man das Tool verwendet.
4. **Flexibilität.** Damit ist die Flexibilität, die man beim Entwickeln einer Benutzeroberfläche zur Verfügung hat, gemeint. Manche Tools geben eine einzuhaltende Methode für die Entwicklung vor, während andere flexibler sind, was natürlich besser ist.
5. **Kommunikation mit anderen Subsystemen.** Die Unterstützung der Kommunikation mit anderen Systemen, wie einer Datenbank oder des Webs, erleichtert die Entwicklung der Benutzeroberfläche.
6. **Erweiterbarkeit und Modularität.** Erweiterbare und modulare, wieder verwendbare Funktionen des Tools sind von Vorteil.

Die Unterstützung der Programmierung für fortgeschrittene Informationsvisualisierungen mit Tools ist essentiell für eine steigende Verwendung der neuen Methoden. Die meisten Tools wurden für eine bestimmte Visualisierungsart entwickelt und können nicht erweitert werden. Im nächsten Teil werden zwei Werkzeugsätze beschrieben, die die Programmierung mehrerer Visualisierungsmethoden ermöglichen.

TOOLKITS

Zur Unterstützung der Programmierung fortgeschrittener Visualisierungsmethoden existieren bereits ein paar Tools. Bekannte Werkzeuge sind PAD++ und Jazz[1], GGobi[2], XML Toolkit[3], Polaris[4], Piccolo[5] und das GeaVista-Studio[6] [3]. Diese Tools eignen sich für unterschiedliche Aufgaben. In diesem Abschnitt wird das Softwaredesign von den zwei Toolkits *InfoVis* und *prefuse* gezeigt. Diese beiden Werkzeuge sind in Java Swing Anwendungen integrierbar, erweiterbar und ermöglichen die Programmierung von mehreren neueren Visualisierungsmethoden. Die Programmgröße der Implementierung von Visualisierungen kann mit diesen Toolkits stark reduziert werden. Für die Darstellung sind nur mehr wenige (hundert) Zeilen Code nötig.

InfoVis Toolkit

Dieses frei erhältliche Toolkit existiert in Form einer Java Library, die es seit 2004 gibt. Das Tool besteht aus einer Softwarearchitektur und Java basierten Komponenten. Unterstützte Visualisierungsmethoden sind ScatterPlots, Zeitreihen, parallele Koordinaten und Matrizen für Tabellen, Treemaps, IcicleTrees und Node-Link-Diagramme für Bäume und Adjazenzmatrizen und Node-Link-Diagramme für Graphen. FisheyeViews und Labeling können auf alle diese Layouts angewendet werden.

InfoVis besitzt folgende Eigenschaften und Funktionen: generische Datenstrukturen, Algorithmen für die Visualisierung dieser Datenstrukturen, direkte Manipulation, Auswahl- und Filtermöglichkeiten, Labeling und räumliche Umformung (deformation) [3].

Architektur

Die Architektur des Tools besteht aus den fünf wichtigen Teilen: Tabellen, Spalten, Visualisierungen, Komponenten und Input/Output. Abbildung 12 zeigt die Architektur des Toolkits mit den einzelnen Bestandteilen. In der Tabelle sind die Daten gespeichert und werden der Visualisierungs-

---

1 http://www.cs.umd.edu/hcil/pad++/
2 http://www.ggobi.org/
3 http://www.thexmltoolkit.org/
4 http://graphics.stanford.edu/projects/polaris/
5 http://www.cs.umd.edu/hcil/jazz/
6 http://www.geovistastudio.psu.edu/jsp/index.jsp

komponente übergeben, die die Daten in eine visuelle Form bringt und dann ein Bild daraus erzeugt. Die Visualisierung kann über dynamische Abfragen und Kontrollen manipuliert werden und somit die Darstellung des Bildes aktualisieren.

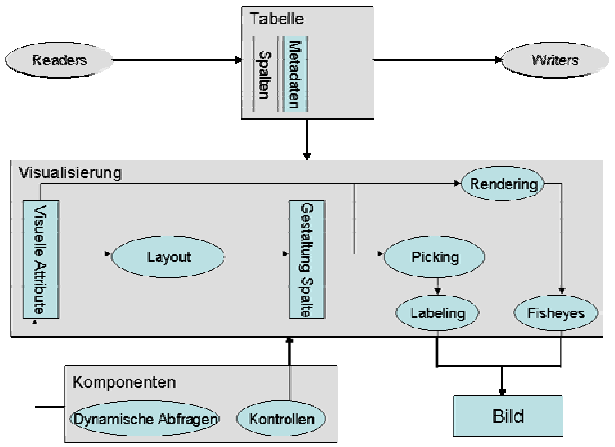


Abbildung 12: Struktur von InfoVis [3],
Vierecke: Datenstrukturen, Ellipsen: Funktionen

Tabelle. Um die Speicherfläche zu verkleinern und die Performance zu erhöhen, wählte man eine einheitliche Datenstruktur in Form von Tabellen. Die gesamte Anwendung baut auf den Tabellen auf. Sie bestehen aus benannten Spalten, Metadaten und Benutzerdaten. Einige Spalten können auch undefiniert sein. Bäume und Graphen sind als Wrapper-Klassen auf den Tabellen implementiert.

Spalten. Spalten beinhalten Java Objekte, abgeleitete Datentypen und im Fall, dass eine Spalte undefiniert ist, einen Null-Wert. Spalten beinhalten außerdem Metadaten, sie können Meldungen auslösen, wenn ihr Inhalt geändert wird und sie unterstützen die Formatierung von Input und Output. Topologische Informationen für Bäume und Graphen wurden als interne Spalte implementiert. Interne Spalten sind Spalten mit einem bestimmten Präfix vor dem Namen und beinhalten interne Daten. Bei einem Baum würde man zum Beispiel die Spalten *parent*, *first* und *next* hinzufügen [3]. Interne Spalten können auch für die Auswahl (boolean Werte) und dynamisches Filtern (Bitmenge) verwendet werden.

Visualisierung. Die Visualisierungskomponente bringt die Attribute der Tabelle in eine visuelle Form. Sie beinhaltet Layout-, Picking- und Renderingfunktionen. Wenn sich Tabellenwerte ändern, wird die Visualisierung neu gestaltet. Jede Visualisierung führt eine Liste von visuellen Attributen. Zu diesen Attributen gehörten Farbe, Größe, Label, Transparenz und Sortierordnung. Es ist auch möglich Visualisierungen zu stapeln, zum Beispiel haben NodeLink Visualisierungen zwei Schichten (eine für die Knoten und die andere für den dazu gehörenden Link). Die Visualisierung besteht aus den Subkomponenten *Colors, Permutations, Redisplay, Rendering, Labeling* und *Spatial Deformations*, die die Darstellung und die Manipulation der Visualisierungen regeln. *Colors* ist für die Farbgebung der Objekte zuständig. *Permutations* braucht man für das Sortieren und das Filtern (Deep-Filtering). *Redisplay* ist zwischen Layout und Rendering aufgeteilt. Diese Subkomponente berechnet das Layout, das die meiste Zeit wieder verwendet werden kann. Weiters passt sie den Bildschirm, aufgrund von Filteroperationen (Auswahl, dynamische Labels), die die Darstellung ändern, an (Rendering). *Rendering* ist für das Zeichnen der Objekte zuständig. Diese Funktion beruht auf Basisformen, Farbberechnungen und FisheyeViews. *Labeling* unterstützt Tooltips oder dynamisches Labeling. *Spatial Deformations* verzerrt die Formen, zum Beispiel damit man einen FisheyeView erhält.

Komponenten. Komponenten regeln die Interaktion mit Interaktionskomponenten. *InfoVis* bietet auch Klassen an, die eine interaktive Manipulation unterstützen.

Dynamische Abfragen. Dynamische Abfragen haben zwei Aufgaben: den Filter für die Visualisierungen und die Java Swing Komponente für die aktuelle Interaktion zu managen.

Beurteilung

Die selbst erteilte Beurteilung (nach den Kriterien des vorherigen Kapitels) des Tools von Jean-Daniel Fekete kommt zu folgender Erkenntnis [3]:

- **Teile der Anwendung:** Datenstrukturen, Präsentationskomponente und Interaktionskomponente sind Teil der Anwendung.
- **Lernzeit:** Die Lernzeit für die Implementierung neuer Visualisierungsmethoden ist lang (Wochen).
- **Entwicklungszeit:** kurz (Stunden)
- **Flexibilität:** Zuerst erstellt man die spezifischen Datenstrukturen, dann wendet (erstellt) man Visualisierungen an, falls notwendig auch Interaktionen, und zum Schluß erstellt man spezifische Kontroll-Panels (wenn nötig).
- **Kommunikation mit anderen Subsystemen:** mehrere Input/Output Formate (CVS, XML,DOT), Java Swing Mechanismen
- **Erweiterbarkeit und Modularität:** sehr erweiterbar aber keine 3D Unterstützung

Code Beispiel

Für die Implementierung fortgeschrittener Implementierungen sind nur mehr wenig LOC erforderlich. In diesem Abschnitt soll Code gezeigt werden (Abbildung 13), der die Darstellung einer TreeMap mit dem *InfoVis* Toolkit implementiert. Dieses Beispiel dient nur zur Veranschaulichung. Man sieht damit, dass der Code nicht lang und leicht zu verstehen ist. Genauere Informationen sind in [14] erhältlich.

```
import infovis.tree.DefaultTree;
import infovis.tree.io.TreeReaderFactory;
import in
fovis.tree.visualization.TreemapVisualization;
import infovis.io.AbstractReader;
import infovis.panel.ControlPanel;
import infovis.panel.ControlPanelFactory;
import javax.swing.JFrame;

public class Example2 {
 public static void main(String[] args) {
  String fileName =
   (args.length == 0) ? "data/salivary.tqd" :
args[0];
   DefaultTree t = new DefaultTree();
   AbstractReader reader =
   TreeReaderFactory.createReader(fileName, t);
   if (reader == null || !reader.load()) {
    System.err.println("cannot load " + fileName);
        }
   TreemapVisualization visualization =
    new TreemapVisualization(t, null, Squari-
fied.SQUARIFIED);
    ControlPanel control =
    ControlPanelFac-
tory.sharedInstance().createControlPanel(
visualization);

   JFrame frame = new JFrame(fileName);
   frame.getContentPane().add(control);
   frame.setVisible(true);
   frame.pack();
    }
}
```

Abbildung 13: *InfoVis* Code für TreeMap [14]

prefuse Toolkit

Dieses Tool wurde 2005 von Jeffrey Heer et al. in [4] vorgestellt. Es ist auch ein Java Library und benutzt weiters die Java2D Grafik Library. *prefuse* beinhaltet verschiedene Layout-Algorithmen, Navigations- und Interaktionstechniken, integrierte Suche und noch mehr. Es unterstützt die Visualisierung von TreeMaps, ScatterPlots, animiertes Radial-Layout und Force-Directed-Layout, FisheyeViews und bifokaler Verzerrung. Da es erweiterbar ist, können weitere Visualisierungsmethoden hinzugefügt werden.

Architektur und Funktionen

Das Framework baut auf dem Referenzmodell des vorherigen Kapitels (siehe Abbildung 11) auf. Abbildung 14 zeigt das Softwaredesign von *prefuse*. Man bringt Daten in eine visuelle Form und leitet daraus Views ab. Durch die Interaktion mit dem Benutzer können sich die unterschiedlichen Transformationsergebnisse (Transformationen sind Filtern und Rendering) ändern, was sich in der Visualisierung zeigt.

Die Eigenschaften und Funktionen der einzelnen Komponenten des Tools, die in [4] erklärt wurden, betrachten wir nun näher.

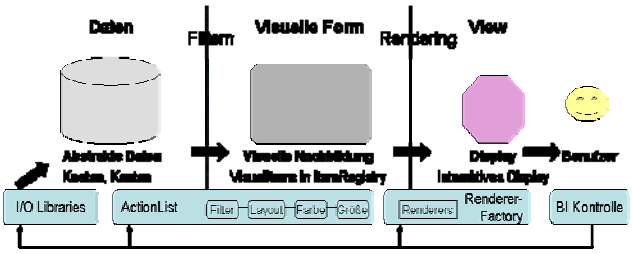


Abbildung 14: prefuse-Design [4]

Daten. Das *prefuse* Toolkit baut auf abstrakten Daten auf. Der elementare Datentyp *Entity* ist die Basisklasse für die Typen *Node*, *TreeNode* und *Edge* und unterstützt Namen-Werte Paare. Eingabe und Ausgabe von Daten werden über ein erweiterbares Interface unterstützt. Es ist auch möglich Daten über Datenbanken oder andere externe Speicher zu erhalten.

Filtern. Mit dem Filtern bringt man die abstrakten Daten in eine für die Visualisierung passende Form. Dafür wählt man die Daten, die dargestellt werden sollen, aus und generiert visuelle Nachbildungen, so genannte *VisualItems*. Diese können zusätzlich Eigenschaften wie Farbe und Größe speichern.

ItemRegistry. Es gibt drei vorgegebene *VisualItems*, die von der *ItemRegistry* gesteuert werden. Das sind die Objekte *NodeItems* (für einzelne Entitäten), *EdgeItems* (für die Relationen) und *AggregateItems* (für Gruppen von Entitäten). Man speichert sie getrennt von den Quelldaten. Die *ItemRegistry* ist eine Datenstruktur, die alle Zustände für eine Visualisierung aufnimmt und außerdem einen *Fokus-Manager* enthält.

Aktionen. Für das Anwendungsdesign verwendet man Aktionen. Sie aktualisieren die *VisualItems* in der ItemRegistry und bieten Funktionen für die Auswahl von visualisierten Daten und das Setzen von visuellen Eigenschaften. Die häufigsten Aktionen sind Filter-, Zuweisungs- und Animatoraktionen. Filteraktionen wurden schon zuvor besprochen. Zuweisungsaktionen setzen Attribute wie Platz, Farbe, Font und Größe von *VisualItems*. Animatoraktionen interpolieren visuelle Attribute (Farbe, Font, …) zwischen Start- und Endwerten, um Animationen auszuführen.

ActionList/Aktivitäten. Für die Datenverarbeitung gibt man die Aktionen in ActionLists, die diese sequentiell ausführen. Sie werden nach Benutzer- oder Systemevents aufgerufen. Man kann sie so konfigurieren, dass sie einmal oder periodisch ausgeführt werden. Die Anwendung der ActionLists wird durch einen Scheduler gemanagt, der in einem Thread läuft.

Rendering. Die VisualItems werden mit Renderers Komponenten auf das Display gezeichnet. Renderers benutzen die visuellen Attribute eines Objekts, um die aktuelle Position der VisualItems zu bestimmen. *prefuse* hat standardmäßig Renderers implementiert, die das Zeichnen von Basisformen, gerade und gewölbte Kanten, Text und Bilder

unterstützt. Die Abbildung von Objekten auf Zeichnungen wird von einer RendererFactory geregelt. Die RendererFactory gibt zu allen VisualItems den entsprechenden Renderer zurück.

Display. Diese Komponente liefert die Präsentation der visualisierten Daten. Display ist eine Subklasse von der Java Swing JComponent Klasse und kann in allen Java Swing Anwendungen verwendet werden. Es beinhaltet eine Liste der sichtbaren Objekte, wendet View-Transformationen an, berechnet die Ausschnittsregion und zeichnet die sichtbaren Objekte mit Hilfe der Renderers. Außerdem sind mehrere Views einer visuellen Form möglich. Tooltips und direkte Manipulation der Objekte sind ausführbar.

Die *prefuse* Library unterstützt die Architektur mit fortgeschrittenen Funktionen, die man für die Visualisierung braucht. Dazu zählen Aktionsmodule für Layout und Verzerrung, Force Simulation, interaktive Kontrollen, dynamic queries, Farbabbildungen, integrierte Suche, Zoomen, *Übersicht + Detail* Anzeige und Event-Logging.

Wenn man das Tool nach den Kriterien in [6] beurteilt, erhält man folgendes Ergebnis:

- **Teile der Anwendung:** Datenstrukturen, Präsentationskomponente und Interaktionskomponente sind Teil der Anwendung.
- **Lernzeit:** Für die Entwicklung von neuen Visualisierungsmethoden wurde keine Lernzeit angegeben.
- **Entwicklungszeit:** kurz (Stunden) [4]
- **Flexibilität:** Die Methode erfordert das Erzeugen von abstrakten Daten (Knoten und Kanten). Die Daten müssen dann in eine visuelle Form gebracht und angezeigt werden. Falls nötig müssen Interaktionen ergänzt werden.
- **Kommunikation mit anderen Subsystemen:** Java Swing, Datenbanken und andere externe Speicher
- **Erweiterbarkeit und Modularität:** sehr erweiterbar und modular aufgebaut, keine 3D Unterstützung

Code Beispiel

Auch für das *prefuse* Toolkit soll eine Beispielcodierung gezeigt werden. Abbildung 15 zeigt Code für die Implementierung einer Netzwerkvisualisierung. Zuerst werden die Daten geladen. Danach wird die Visualisierung für die Daten gebildet. Renderer werden gemacht und der Visualisierung zugeordnet. Dann werden Aktionen, Display und interaktive Kontrollen kreiert. Zum Schluss wird die Visualisierung gestartet. Das Code Beispiel soll nur darauf hinweisen, dass die Implementierung nicht zu kompliziert ist. Genauere Informationen findet man in [7].

```
Graph graph = null;
graph = new GraphMLRea-
der().readGraph("/socialnet.xml");
Visualization vis = new Visualization();
vis.add("graph", graph);

LabelRenderer r = new LabelRenderer("name");
r.setRoundedCorner(8, 8); // round the corners
vis.setRendererFactory(new DefaultRendererFac-
tory(r));

int[] palette = new int[] {
    ColorLib.rgb(255,180,180), Color-
Lib.rgb(190,190,255)
DataColorAction fill = new DataColorAc-
tion("graph.nodes", "gender",
    Constants.NOMINAL, VisualItem.FILLCOLOR, pal-
ette);
ColorAction text = new ColorAction("graph.nodes",
    VisualItem.TEXTCOLOR, ColorLib.gray(0));
ColorAction edges = new ColorAction("graph.edges",
    VisualItem.STROKECOLOR, ColorLib.gray(200));
ActionList color = new ActionList();
color.add(fill);
color.add(text);
color.add(edges);

ActionList layout = new Ac-
tionList(Activity.INFINITY);
layout.add(new ForceDirectedLayout("graph"));
layout.add(new RepaintAction());
vis.putAction("color", color);
vis.putAction("layout", layout);

Display display = new Display(vis);
display.setSize(720, 500);
display.addControlListener(new DragControl());
display.addControlListener(new PanControl());
display.addControlListener(new ZoomControl());

JFrame frame = new JFrame("prefuse example");
fra-
me.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.add(display);
frame.pack();           frame.setVisible(true);
vis.run("color");
vis.run("layout");
```

Abbildung 15: *prefuse* Code für Netzwerk [7]

ZUSAMMENFASSUNG

In dieser Arbeit wurde die Idee einiger fortgeschrittener Visualisierungsmethoden beschrieben. Es wurden Herausforderungen, die beim Entwickeln von Layouts und Visualisierungsanwendungen entstehen, erörtert. Hauptsächlich wurde dabei das Softwaredesign solcher Anwendungen betrachtet. Das Problem vieler Methoden ist, dass die Systeme, die die Implementierung der Layouts unterstützen, nicht erweiterbar und flexibel genug sind. Die Entwicklung eines Tools, das die gesamten Anforderungen erfüllt, erfordert hohe mathematische und programmiertechnische Fähigkeiten. Vor allem die laufende Interaktion mit dem Benutzer, die Reorganisation des Bildschirms und das Zusammenarbeiten mit anderen Systemen erfordern flexible Anwendungen, die nicht leicht zu konstruieren sind. Ein weiterer bedeutender Punkt sind die Benutzerpräferenzen. Damit neue Darstellungsarten angewendet werden, muss

der Benutzer sie akzeptieren. Das fordert eine leichte Lernbarkeit des Layouts und ein rasches Feedback.

Es wurden weiters zwei Tookits vorgestellt, die mehrere Visualisierungen unterstützen und weitere Layouts hinzunehmen können. Ich denke, solche Werkzeuge bieten einen guten Ansatz, damit künftig mehr neue Visualsierungsmethoden für die Darstellung großer Datenmengen angewendet werden.

Referenzen:

1. Kobsa A., "User Experiments with Tree Visualization Systems", InfoVis'04, 2004, pp. 9-16
2. Card S. K., Nation D., "Degree-of-Interest-Trees: A Component of an Attention-Reactive User Interface", AVI2002, 2002, pp. 231-245
3. Fekete JD., „The InfoVis Toolkit", InfoVis'04, IEEE Press, 2004, pp. 167-174
4. Heer J., Card S. K., Landay J. A., "prefuse: A Toolkit for Interactive information Visualization", in Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'05), 2005, pp. 2-7
5. Shneiderman B., Plaisant C., "Designing the User Interface", Addison-Wesley Verlag, 2005
6. Card S. K., Mackinlay J. D., Shneiderman B., "Readings in Information Visualization: Using Vision to Think", Morgan Kaufmann Verlag, 1999
7. Heer J. et al., prefuse Toolkit, Projektseite, 2007 http://prefuse.org/, letzter Besuch: 12. Juni 2007
8. Barlow T., Neville P., "A Comparison of 2-D Visualizations for Hierarchies", InfoVis'01, 2001, pp. 131-138
9. Herman I., Melancon G., Marshall M. S., "Graph Visualization and Navigation in Information Visualization: A Survey", IEEE Transactions on Visualizations and Computer Graphics, vol. 6, no. 1, 2000, pp. 24-43
10. Plaisant C., Grosjean J., Bederson B. B., "SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation", InfoVis'02, 2002, pp. 57-64
11. Plaisant C. et al., SpaceTree Projekt, 2002, http://www.cs.umd.edu/hcil/spacetree/, letzter Besuch: 11. Juni 2007
12. Lamping J., Rao R., "Visualizing Large Trees Using the Hyperbolic Browser", Conference on Human Factors in Computing Systems, 1996, pp. 388-389
13. Heer J., Information Visualization Infrastructure, 2004, http://vw.indiana.edu/ivsi2004/jherr/index.html, letzter Besuch: 12. Juni 2007
14. Fekete JD., InfoVis Toolkit, Projektseite, 2007, http://ivtk.sourceforge.net/, letzter Besuch: 12. Juni 2007
15. Shneiderman B., „Tree Visualization with Tree-Maps: 2-d Space-Filling Approach", ACM Transactions on Graphics, vol. 11., no. 1, 1992, pp. 92 – 99
16. Ahlberg C., Shneiderman B., "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays", Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence, 1994, pp. 313-317
17. Sarkar M., Brown M. H., "Graphical Fisheye Views of Graphs", Proceedings of the SIGCHI conference on Human factors in computing systems, 1992, pp. 83- 91
18. Leung Y. K., Apperley M. D., „A Review and Taxanomy of Distortion-Oriented Presentation Techniques",ACM Transactions on Computer-Human Interaction (TOCHI), vol. 1, no. 2, 1994, pp. 16-160

3D-Dateiverwaltungssysteme

Stefan Urabl
0360329
stefan.urabl@web.de

ZUSAMMENFASSUNG
Diese Arbeit beschäftigt sich mit 3D-GUIs zur Datei- und Ordnerverwaltung und versucht zu ergründen, warum sich die Verwendung der dritten Dimension in diesem Arbeitsbereich nicht durchgesetzt hat. Es werden die grundlegenden Techniken zur 3D-Datenvisualisierung vorgestellt sowie einige Beispiele aus der Praxis mit ihren Vor- und Nachteilen betrachtet. Außerdem wird ein kurzer Blick auf die Verwendung von 3D-Oberflächen als Internet-Browser geworfen. Abschließend wird nach effizienteren Anwendungsmöglichkeiten für 3D-Dateiverwaltungssysteme gesucht.

Nicht betrachtet wurden 3D-Desktops, die keine Abbildung der Ordnerstruktur, sondern eine 3D-Umsetzung eines Symbole-und-Oberflächen-Systems sind.

Keywords
3D, Dateisystem, Ordnerverwaltung, Dateiverwaltung, Cone-Tree, Tree-Map

1. EINFÜHRUNG

Der Umfang und die Komplexität der Daten auf Computersystemen wächst stetig. Die Software zur Verwaltung dieser Daten verändert sich aber nur langsam. Heute finden sich wie schon seit Jahren meist zweidimensionale grafische Oberfläche, basierend auf Icons, Listen und Bäumen, mit jeder Betriebssystem-Generation nur um unwesentliche Details erweitert.

Die Verwendung der dritten Dimension zur Visualisierung von Dateien, Ordnern und deren Beziehungen ist immer noch ungewöhnlich, sorgt immer wieder für beeindruckte und neugierige Blicke und findet sich vor allem in eher futuristischen Szenarien und Filmen (Zum Beispiel Jurassic Park). Inzwischen haben viele solcher Systeme schon einen sehr fortgeschrittenen Entwicklungsstand erreicht und sind oft auch frei verfügbar, trotzdem haben sich 3D-Systeme zur Dateiverwaltung nie von einer gelegentlichen Spielerei weg und hin zu einer verbreiteten Anwendung entwickeln können.

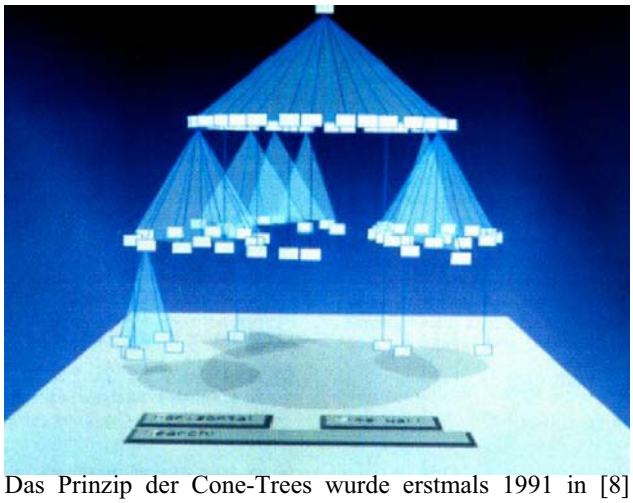
Mehrere Studien versuchten bereits Anwendungsgebiete für 3D-Darstellungen zu finden, die eine effizientere Informationsaufnahme ermöglichen würden. So untersuchten [4] und [5] ob eine dreidimensionale Anordnung der Elemente (perspektivische Verzerrungen und Schattenwürfe) die Orientierung und die Merkfähigkeit der Testpersonen steigern könnte. Die 3D-Darstellungen konnten allerdings keine merkliche Effizienzsteigerung erzielen, sondern blieben bei den meisten Aufgaben sogar hinter die 2D-Darstellungen zurück. Allerdings gaben viele Testpersonen an, sich mit der 3D-Darstellung wohler zu fühlen. 3D-Verwaltungssysteme sollten also einer näheren Betrachtung unterzogen werden.

2. TECHNIKEN ZUR DREIDIMENSIONALEN DATENVISUALISIERUNG

Im Laufe der Jahre sind viele Techniken aufgekommen, um Information dreidimensional darzustellen. Im folgenden soll auf die verbreitetsten dreidimensionalen Visualisierungen für hierarchische Strukturen (Zum Beispiel von Ordnern mit Unterordnern und Dateien) eingegangen werden.

2.1 Cone-Trees

Bäume sind schon seit langem eine verbreitete Darstellungsform für hierarchische Strukturen. Cone-Trees erweitern dieses Konzept in den dreidimensionalen Raum.



Das Prinzip der Cone-Trees wurde erstmals 1991 in [8] vorgestellt.

Während in einem zweidimensionalen Baum die Kind-Knoten in einer Linie unter dem Elternknoten aufgereiht werden, werden sie in Cone-Trees in Form eines rotierbaren Kreises angezeigt. Der offensichtlichste Nutzen einer solchen Präsentationsform ist Platzgewinn. Ein breiter

Baum ist in zweidimensionaler Darstellungsform oftmals nicht zur Gänze auf dem Bildschirm sichtbar.

Allerdings werden auch Cone-Trees ab einer gewissen Anzahl von Elementen unübersichtlich. ([8] nennt hier ungefähr 10000 Knoten oder 10 Ebenen).

[8] sieht Cone-Trees als am effizientesten bei der Darstellung unbalancierter Hierarchien. Balancierte Bäume mit vielen Elementen führen schnell zu Überdeckungseffekten, einige Zweige werden also durch andere verdeckt.

[3] beschreibt eine Studie zur Effizienz von Cone-Trees.

Dabei wurden Cone-Trees mit äquivalenten, flachen Bäumen hinsichtlich der Ausführungsgeschwindigkeit bestimmter Aufgaben verglichen. Die Aufgaben variierten in drei Eigenschaften: Depth (Tiefe der Suchaufgabe. Mögliche Werte waren Shallow=2 und Deep=4), Density (Anzahl der Unterdirectories, also Breite des Baums. Mögliche Werte waren Sparse=6, Medium=10 und Dense=20). Die Aufgaben (Tasks) 1-6 starten von der Wurzel eines minimierten Baums und expandieren diesen während eines Suchvorgangs. Aufgabe 7 stellt die Rückkehr vom expandierten zum minimalisierten Baum da und ist daher von Aufgabe 6 abhängig.

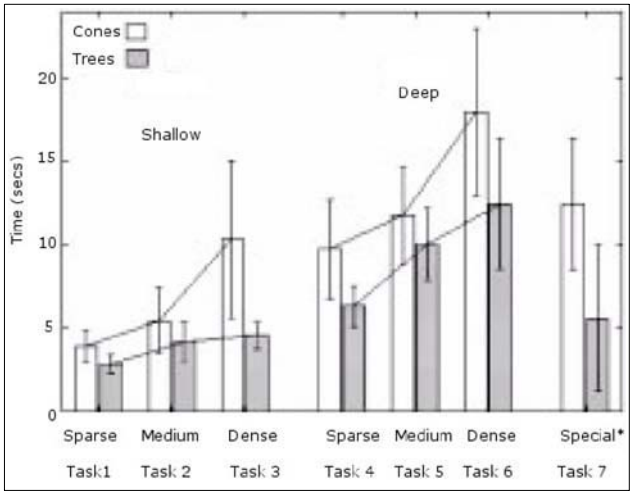


Abb. 2 Cone-Tree-Evaluierung [3, Texte zur besseren Lesbarkeit bearbeitet]*

Wie aus Abb. 2 erkennbar, war die Verwendung der Cone-Trees in jedem Bereich langsamer als mit den üblichen flachen Bäumen. Auch die Anmerkungen der Testpersonen bezüglich Übersichtlichkeit und Effektivität der Cone-Tree-Implementierung fielen zu deren Ungunsten aus.

* Bei Task 7 war die Beschriftung im Original-Dokument nicht zu entziffern. Wie aus dem Text zu entnehmen ergibt sich die Tiefe von Task 7 allerdings aus den vorhergehenden Tasks.

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7
			"It was easy to see the data needed"				
Cone: Mean(st.dev.)	3.92(0.9)	3.42(0.79)	2.42(1.0)	3.75(0.97)	3.08(0.9)	2.67(1.5)	3.17(1.47)
Tree: Mean(st.dev.)	4.58(0.51)	4.5(0.52)	4.08(0.9)	4.5(0.52)	3.92(0.67)	3.83(0.84)	4.33(0.49)
Paired T(11)	-2.35	-4.17	-4.43	-2.69	-3.08	-2.55	-2.76
p	0.039	0.002	0.001	0.021	0.01	0.03	0.019
			"Overall the interface was effective for the task"				
Cone: Mean(st.dev.)	4.17(0.72)	3.67(0.78)	2.83(0.94)	3.67(0.98)	3.33(0.78)	3.0(1.54)	3.17(1.34)
Tree: Mean(st.dev.)	4.75(0.45)	4.67(0.49)	4.17(0.83)	4.58(0.52)	3.83(0.72)	3.75(0.45)	4.25(0.45)
Paired T(11)	-2.55	-3.63	-4.0	-2.93	-2.17	-1.83	-2.6
p	0.027	0.004	0.002	0.014	0.053*	0.095*	0.025

Abb. 3 Bewertungen der Testpersonen [3]

Die Autoren geben aber zu bedenken, dass die meisten Testpersonen zuvor niemals mit Cone-Trees zu tun hatten, während sie allerdings durchaus Erfahrung mit flachen Bäumen aufwiesen. Der Test war also nicht zur Gänze neutral.

Manche Testpersonen waren außerdem trotz der schlechteren Effizienz begeistert von den Cone-Trees und gaben an, dass sie eine besseres Gefühl für die Datenstruktur vermitteln. Möglicherweise ist hier mit besseren Navigationsmethoden noch eine Verbesserung zu erreichen.

2.2 Space-filling tree-maps

Tree-maps verwenden eine rechteckige Darstellung von Knoten. Im Gegensatz zur verbreiteten Baumstruktur sind sie mit ihren Kind-Knoten nicht durch Kanten verbunden, sondern diese werden als kleinere Rechtecke in den Eltern-Knoten eingebettet.

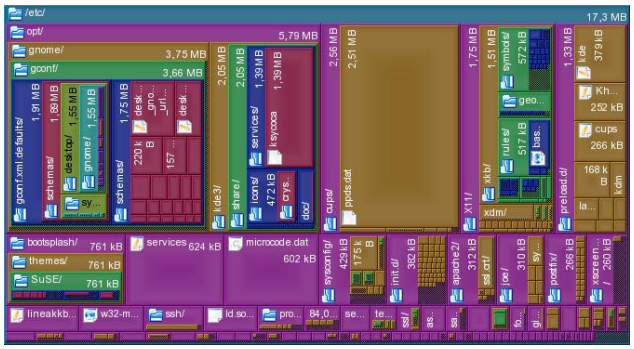


Abb. 4 2D-Tree-map [KDE-Konqueror]

Die Fläche der einzelnen Knoten ist proportional zu einer spezifischen Eigenschaft (Zum Beispiel der Dateigröße beim Einsatz als Dateiverwaltungssystem).

3D-Space-filling-tree-maps erscheinen als Blockstapel oder Stufenpyramiden. In manchen Implementierungen dient auch die Blockhöhe als Maß und die Grundfläche bleibt konstant. Der Vorteil der 3D-Anzeigeform ist, dass bei tiefen Hierarchien die Basisknoten (Grund-Directories) leichter zu erkennen sind (Für veranschaulichende Bilder siehe 3.1).

[9] vergleicht 2D- und 3D-Implementierungen anhand der Beispielimplementierungen TreeMap und StepTree (3.1).

Die Testpersonen hatten neun Aufgaben mit beiden Anzeigeformen durchzuführen:

1. Finde die größte Datei.

2. Finde die größte Datei eines speziellen Dateityps.
3. Finde das Verzeichnis, das am weitesten unten in der Hierarchie liegt.
4. Finde eine Datei in einem bestimmten Pfad.
5. Finde heraus, welches von zwei Verzeichnissen die meisten Dateien und Sub-Verzeichnissen beinhaltet.
6. Welches von zwei Verzeichnissen ist größer?
7. Welcher ist der verbreitetste Dateityp?
8. Finde heraus in welchem Verzeichnis eine aktuell ausgewählte Datei sich befindet.
9. a) Finde die größte Datei in einem Verzeichnis

   b) Finde die größte Datei des selben Typs in der gesamten Hierarchie

Abb. 4 zeigt, dass die 3D-Implementierung StepTree in den meisten Bereichen äquivalent zur 2D-Implementierung TreeMap war, Aufgabe 3 (Finde das Verzeichnis, das am weitesten unten in der Hierarchie liegt) jedoch erheblich schneller und mit geringerem Fehlergrad erfüllt wurde.

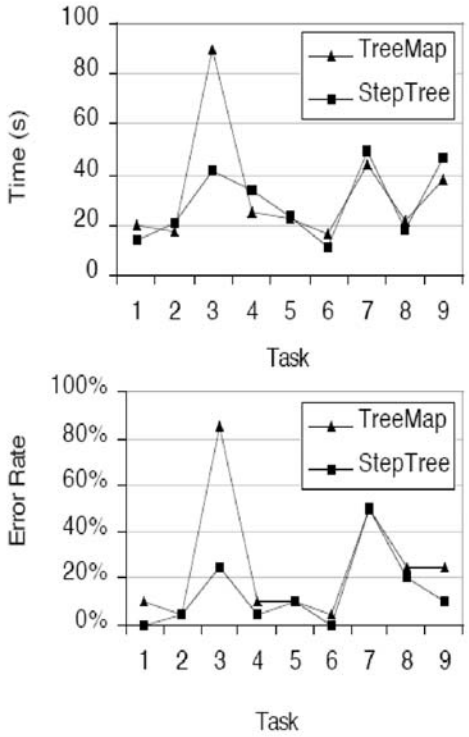


Abb. 4 Vergleich TreeMap (2D) und StepTree (3D) [9]

Die Verwendung der dritten Dimension führte hier also tatsächlich zu einer Erhöhung der Übersichtlichkeit im Bezug auf die hierarchischen Strukturen.

Einige dieser Aufgaben sind in üblichen Baumstrukturen nicht oder nur unter großem Aufwand durchzuführen. Leider konnte aber kein experimenteller Vergleich zwischen 3D-Space-filling-tree-maps und klassischen Baumstrukturen gefunden werden.

2.3 Verschachtelte Objekte

Eine weitere Präsentationsform stellt Directories als Würfel oder Kugel dar, in die weitere gleichartige Objekte (Die Subdirectories) und die Dateien eingebettet sind. Die Navigation erfolgt hier meist wie in einem Computerspiel, man kann also mit Maus und Tastatur in der virtuellen Welt spazieren oder fliegen.

Während diese Directory-Welten meist sehr eindrucksvoll wirken, sind zumindest die bisherigen Implementierungen für den normalen Gebrauch relativ unhandlich. Da man sich „innerhalb“ eines Verzeichnisses befindet, sind Verzeichnisse oder Dateien, die in der Hierarchie höher oder mehr als eine Ebene tiefer stehen, meist nicht oder nur teilweise einsehbar.

Sie sind allerdings hervorragend dazu geeignet, sich oder anderen ein Bild von der Verzeichnisstruktur des darunter liegenden Systems zu machen.

Beispiele sind 3.2, 3.2, 3.4.

3. BEISPIEL-REALISIERUNGEN

Im folgenden werden einige aktuelle Programme vorgestellt, die eine oder mehrere der zuvor beschriebenen Techniken nutzen um, je nach Programm vollständige oder eingeschränkte, Dateiverwaltung zu ermöglichen. Dabei wird vor allem auf die positiven und negativen Eigenschaften eingegangen, die sich während der Tests gezeigt haben. Die hier angeführten Beispiele wurden alle nur vom Author selbst getestet, da von den meisten zwar Beschreibungen, aber keine vergleichenden Experimente vorliegen. StepTree wurde allerdings in [9] analysiert (Siehe dazu auch 2.2).

Letztendlich wird dadurch versucht Vor- und Nachteile und sinnvolle Einsatzgebiete von 3D-Dateiverwaltungssystemen zu ermitteln.

Einige weitere Beispiele finden sie in [1].

3.1 StepTree

StepTree ist eine Freeware-Implementierung der Space-filling tree-maps für Windows.

Das Dateisystem erscheint als schräge Draufsicht. Dateien und Ordner werden als Quader dargestellt und abhängig von ihrem Typ eingefärbt. Dateien und Unterordner werden dabei auf ihren Elternordner gestapelt. Die Grundfläche der Quader ist proportional zur Größe der Datei beziehungsweise des Ordners. Zusätzlich wird außerdem der Änderungszustand dargestellt: Dateien die in den letzten drei Jahren geändert wurden, sind solide, ebenso Ordner, die solche Dateien beinhalten. Alle anderen (Also

Dateien und Ordner die lange nicht geändert wurden) werden transparent dargestellt.

Dadurch sind Dateien mit bestimmten Eigenschaften (Lange nicht geändert, groß, bestimmter Dateityp) leichter über Hierarchie-Grenzen hinaus erkennbar als in Baumstrukturen. Allerdings sind exakte Eigenschaften (Genaue Dateigröße, Dateinamen) ohne zu zoomen nur schwer erkennbar.

Beim Start des Programs wird ein Einstiegsordner gewählt, der dann präsentiert wird.

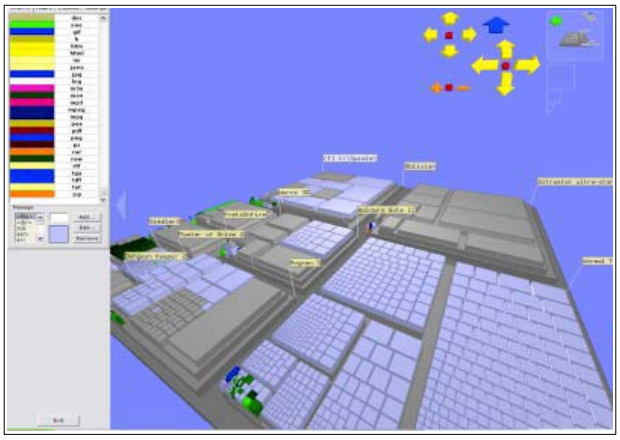


Abb. 5 StepTree

Die Navigation erfolgt über Rotation und Zooming, durch Doppelklick auf einen Ordner, kann dieser zum neuen Hauptansichts-Ordner gemacht (Also in den Vordergrund geholt) werden.

StepTree bietet keine Manipulationen der Dateien, sondern nur eine Ansicht. Diese Aufgabe erfüllt es aber besser als die meisten anderen 3D-Dateiverwaltungssysteme, wird aber ebenso bei einer größeren Anzahl von Dateien unübersichtlich (Siehe Abb. 4).

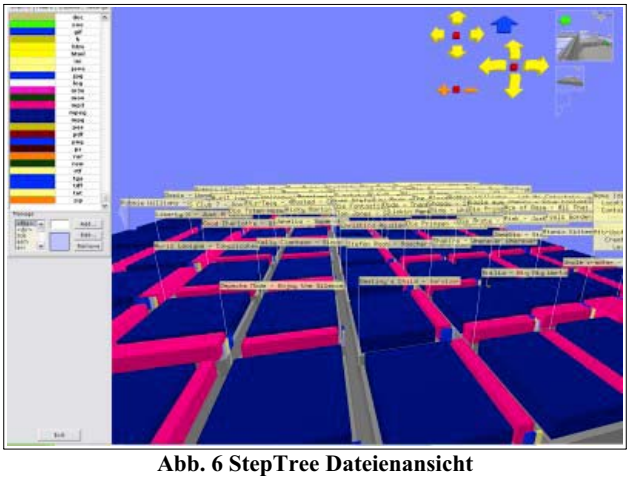


Abb. 6 StepTree Dateienansicht

Wirklich brauchbar ist StepTree eigentlich nur für die Suche nach großen Dateien oder Ordnern (Wenn zum Beispiel der Festplattenplatz knapp wird). Diesen Zweck würde aber eine 2D-Entsprechung (Wie sie sich z.B. in KDE findet) auch erfüllen.

[2] Beschreibt StepTree im Detail.

3.2 Tactile

Tactile ist eine kommerzielle Software für Windows. Für diese Arbeit wurde die freie Trial-Version herangezogen.

Tactile verwendet den Ansatz der verschachtelten Objekte (2.3) und stellt Ordner als Kugeln dar, die wiederum gleichartige kleinere Kugeln (die Unterordner) und verschiedene Objekte (die Dateien) beinhalten.

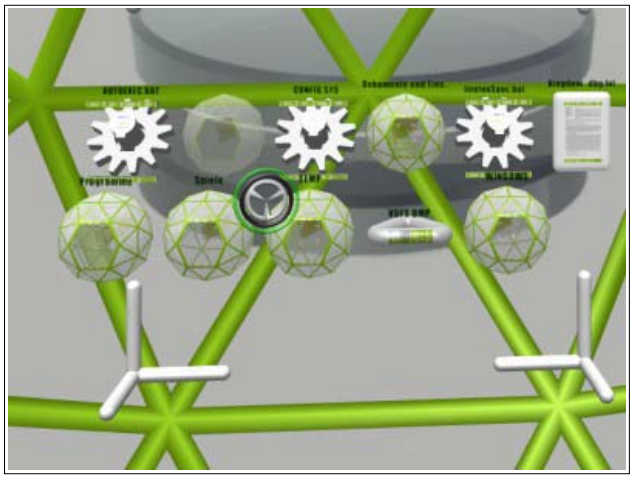


Abb. 7 Tactile Ordner und Dateien

Die Navigation erfolgt ähnlich wie bei vielen 3D-Computerspielen, die Blickrichtung wird mit der Maus festgelegt, die Bewegung mit den Cursortasten und dem Mouse-Wheel gesteuert. Man kann in die Ordner-Kugeln “fliegen” und so zwischen den Ordnern wechseln. Etwas schneller kann man sich einem Ordner durch Doppklick annähern.

Dateien werden entsprechend ihrer Art als verschiedene 3D-Objekte dargestellt. Bilddateien zum Beispiel als freischwebende Mini-Bilder, die sich immer auf den Benutzer ausrichten. Positioniert werden sie in mehreren übereinander liegenden und versetzten Reihen, die Dateiinformationen (Name, Größe,...) schweben in der Nähe und werden bei Annäherung sichtbar. Die genaue Darstellungsform lässt sich durch auswechselbare Themes variieren.

Tactile unterstützt auch die üblichen Dateioperationen (Löschen, Kopieren,...), indem bei Auswahl einer Datei entsprechende Kontextmenüs eingeblendet werden können. Drag & Drop ist ebenfalls möglich, nachdem bei einem Objekt „Grab“ ausgewählt wurde, folgt es den Bewegungen des Benutzers bis es fallengelassen wird („Drop“).

Die im Test interessanteste Anwendungsmöglichkeit von Tactile war die Bildergalerie. Man kann in einem Ordner mit Bilddateien von Bild zu Bild fliegen und sich

interessante Bilder näher ansehen (Abb. 8). Allerdings werden die Grafiken auch bei maximaler Annäherung nicht in voller Qualität angezeigt. Eine entsprechende Überarbeitung würde diesen Teil der Software deutlich interessanter machen.

Abb. 8 Bilddateien in Tactile

Letztendlich ist aber auch Tactile zu umständlich und es ließen sich keine bemerkenswerten Vorteile gegenüber einer 2D-Anwendung erkennen. Das Herumfliegen in den eigenen Ordnern macht einige Minuten Spaß, zur tatsächlichen Verwendung als Dateisystem ist Tactile aber nicht geeignet. Die Bewegung zu einer bestimmten Datei dauert einfach zu lange (Der Wechsel zu einem zwei Ebenen höher liegenden Ordner, der nicht der Elternordner war, dauerte im Test 4 Sekunden, noch länger wenn man sich im Ordner bewegt hatte, da man dann manchmal am „Eingang“ des Ordners hängen blieb.)

Da immer nur ein Ordner mit seinen Unterordnern sichtbar ist, aber ein Zurückblicken auf höhere Ebenen oder ein klarer Blick auf tiefere nicht machbar ist, lässt auch die Übersichtlichkeit stark zu wünschen übrig.

3.3 Xcruiser

Xcruiser ist ein freies System für Linux und verfolgt einen ähnlichen Ansatz wie Tactile, allerdings auf nicht ganz so hohem grafischem Niveau. Die Steuerung kann zwischen Rotation und Flugbewegung gewechselt werden.

Ordner werden hier als Galaxien dargestellt, Dateien als Sterne, deren Größe abhängig von der Größe der jeweiligen Datei ist. Die Farbe und Position ergeben sich aus dem Dateinamen, daher liegen Dateien mit ähnlichem Dateinamen eng beieinander, was gänzlich neue Suchmethoden eröffnet. Eine interessante Eigenheit von Xcruiser ist außerdem, dass die in Linux-Systemen recht häufigen Hard- und Softlinks als “Wurmlöcher” zwischen einzelnen Ordnern oder Dateien dargestellt werden (in Abb. 9 als grüne Linien erkennbar).

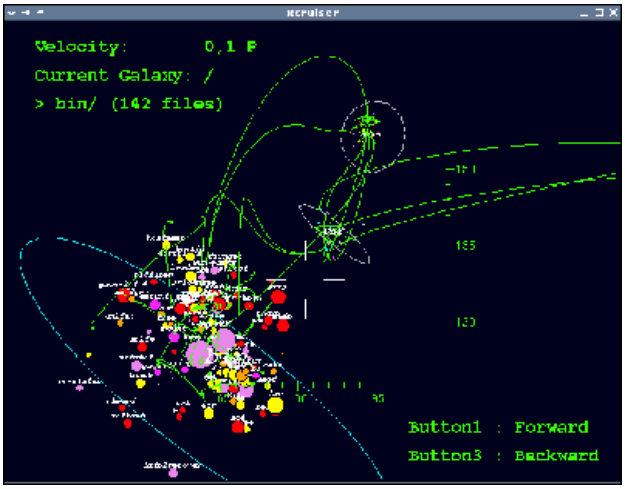


Abb. 9 Xcruiser Links

Dadurch lässt sich Xcruiser gut zur Erklärung für die Linux-Dateistruktur und das Link-Konzept nutzen. Für die praktische Arbeit ist es aber genauso umständlich wie Tactile.

3.4 tdfsb

tdfsb ist im Moment das wahrscheinlich am weitesten entwickelte und umfangreichste 3D-Dateiverwaltungssystem. Es ist ein freies System für Linux.

tdfsb stellt die Objekte des Dateisystems auf einer ebenen Fläche dar. Ordner sind als Blöcke dargestellt, die weitere Objekte (Unterordner oder Dateien) enthalten. Dabei werden Farbcodierungen verwendet um bestimmte Eigenschaften (schreibgeschützt, Verknüpfung,...) zu kennzeichnen. Die Größe der Objekte ist proportional zur Dateigröße.

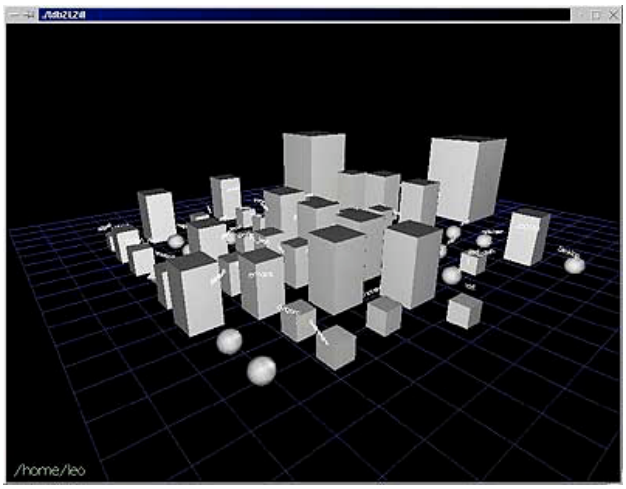


Abb. 10 tdfsb

Interessant ist die Darstellung von Bildern und Soundfiles. Erstere stehen im Raum und lassen sich wie eine Bildergalerie durchwandern, letztere sind als

dreidimensionale Lautsprecher dargestellt. Wenn sie gerade abgespielt wird, vibriert der Lautsprecher.

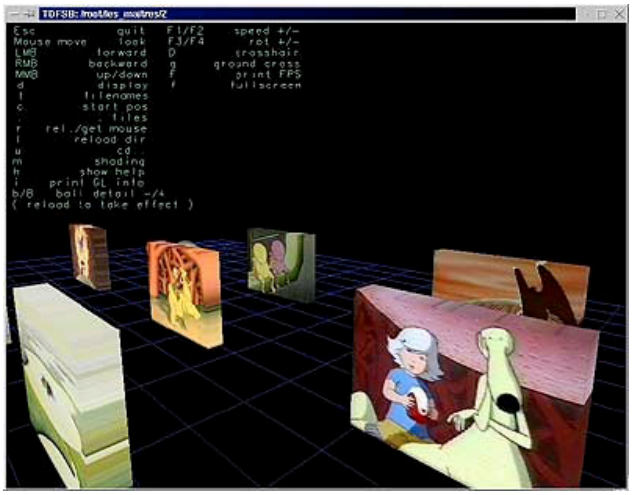


Abb. 11 Bilddateien in tdfsb

Auch für tdfsb gilt aber leider: Der Wow-Effekt überwiegt den praktischen Nutzen bei Standardoperationen. Die Ansicht kann aber durchaus genutzt werden, um eine Übersicht über das Dateisystem zu gewinnen.

4. 3D im Web

Durch die 3D-Darstellung von Verknüpfungen in Xcruiser (3.3) drängt sich die Idee des Einsatzes von 3D-Oberflächen für Internetbrowsing auf. Während der Arbeit an diesem Artikel, ergab sich der Gedanke an eine dreidimensionale Fensterdarstellung, bei der verlinkte Seiten im Hintergrund dargestellt werden, mit der Ursprungsseite allerdings eindeutig verbunden. Auch die Möglichkeit noch gar nicht geöffnete, aber über die Links der aktuellen Seite erreichbare Seiten blass oder transparent als Vorschau im Hintergrund zu öffnen, würde sich anbieten.

Die Nachforschungen in dieser Richtung verliefen aber eher enttäuschend. Es gibt tatsächlich schon einige Ansätze für 3D-Browser, die aber den Erwartungen nicht gerecht wurden, wie die folgenden Beispiele zeigen.

4.1 Einige Beispiele für aktuelle "3D-Browser"

SphereXPlorer [14]

SphereXPlorer ist eine Erweiterung des 3D-Desktops SphereXP [14], der den flachen Windows-Desktop durch die Innenseite einer Kugel ersetzt. Mit SphereXPlorer können Browser-Fenster frei in dieser Kugel platziert und rotiert werden.

Dadurch können theoretisch mehr offene Fenster zugleich sichtbar sein als auf einem üblichen 2D-Desktop.

In der Praxis ist das ständige drehen und Zoomen aber deutlich aufwendiger als die in heutigen Browsern üblichen Tabs.

Abb. 12 SphereXPlorer

Vorschaufunktionen, Verknüpfungsanzeigen oder irgendwelche andere Ergänzungen, die über die Plazierung der Fenster hinausgehen, sind außerdem nicht vorhanden.

LG3D [16]

Looking Glas LG3D ist eigentlich eine komplette 3D-Desktopumgebung von Sun, bietet dabei aber auch genauso wie SphereXPlorer die schräge Positionierung von Browser-Fenstern an. Als Vorteil sind die Plattformunabhängigkeit (da es auf Java basiert) und die Anzeige des Anzeige des Seitennamens an der Seitenkante gedrehter Fenster (Abb. 13) zu nennen.

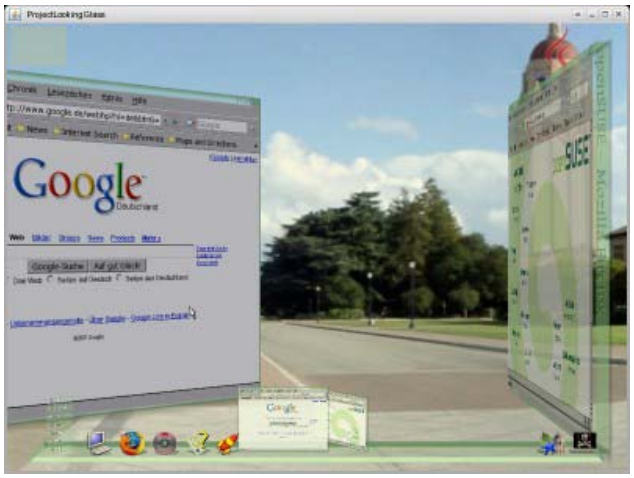


Abb. 13 Project Looking Glass - LG3D

Aber auch hier halten sich die Neuerungen ansonsten in Grenzen. Eine Einblendung von Verknüpfungen als Verbindungen zwischen den Fenstern oder Vorschauen von noch nicht geöffneten Verknüpfungen im Hintergrund würden eine Nutzung der dritten Dimension rechtfertigen und den Nutzen gewaltig steigern. Im Moment überwiegt aber die Umständlichkeit der Bedienung den geringen Nutzen durch den Raumgewinn bei Weitem.

SpaceTime [17]
Am ehesten von allen getesteten Browsern erfüllte noch SpaceTime diese Anforderungen.

SpaceTime zeigt zwar keine noch nicht geöffneten Links an, ermöglicht es aber Suchergebnisse (zum Beispiel von Ebay oder Google) im Hintergrund als Vorschaufenster zu öffnen und mit dem Mouse-Wheel zwischen ihnen zu wechseln oder sie per Doppelklick zu öffnen.

Abb. 14. SpaceTime

Dadurch kann man quasi „im Vorbeigehen" einen kurzen Blick auf die entsprechenden Seiten werfen, was deutlich ergiebiger als die Schlagwörter oder Textauszüge üblicher Suchmaschinen ist. Da sich SpaceTime noch in der Entwicklung befindet ist hier möglicherweise wirklich ein Durchbruch zu einer sinnvollen Nutzung der 3D-Technik zu erhoffen.

Weitere getestete 3D-Browser sind Clara [13], der auch nur eine erweiterte Platzierung von Seiten in einem dreidimensionalen Raum zur Verfügung stellt und 3B [18], der aber bereits den Bereich der virtuellen Welten streift und damit das Thema dieser Arbeit verlässt.

5. SCHLUSSFOLGERUNGEN

Alle gefundenen Benutzertests und auch die Softwaretests des Authors fielen zu Ungunsten der 3D-Umgebungen aus, auch wenn einige durchaus Potential für bestimmte Aufgabenbereiche aufweisen.

Das offensichtlichste Problem vieler 3D-Dateiverwaltungssysteme ist der Versuch, gewohnte 2D-Mechanismen vollständig in den dreidimensionalen Raum zu adaptieren und die entsprechenden zweidimensionalen Applikationen somit zu ersetzen. Eine solche Informations- und Funktionsvielfalt führt aber bei 3D-Anwendungen schnell zur Unübersichtlichkeit.

Die 3D-Dateiverwaltungssysteme wirken daher zwar oft natürlicher und eindrucksvoller als die entsprechenden 2D-Äquivalente, sind aber im Allgemeinen deutlich ineffizienter. Im täglichen Gebrauch ist also der spartanischeren 2D-Darstellung der Vorzug zu geben.

Die meisten der getesteten Programme und Ansätze waren aber zumindest in einem Teilbereich ausgesprochen nützlich, so ist etwa die Suche nach Ordnern und Dateien mit besonders hohem Platzverbrauch mit Space-filling tree-maps wie in StepTree deutlich einfacher und auch das Spazieren durch die Bildergalerien von tdfsb erfolgt weitaus ungezwungener als das übliche Listen-Scrolling.

Das Schlagwort heißt also Spezialisierung. Anstatt die gewohnten 2D-Oberflächen in allen Bereichen zu ersetzen, sollte man die Möglichkeiten von 3D-Darstellungen in bestimmten Bereichen nutzen um diese zu ergänzen.

Experimente mit Oberflächen, die sich übergangslos zwischen 2D- und 3D-Ansicht umschalten lassen, wären wünschenswert.

Eine weitere Anwendungsmöglichkeit wäre der Einsatz zur Erklärung von Dateisystemen im Unterricht, da 3D-Dateiverwaltungssysteme manche Strukturen und Eigenschaften klarer visualisieren.

6. EIN NACHWORT ZUM AKTUELLEN FORSCHUNGSSTAND UND DER ENTWICKLUNG DIESES TEXTES

In den 90er-Jahren haben sich einige Grundkonzepte entwickelt (2), die seitdem kaum mehr eine Änderung durchlaufen haben. Das Problem der Stagnation, das in der Einleitung bereits bei den 2D-Dateiverwaltungssysteme kritisiert wurde, hat also auch bereits die 3D-Systeme erfasst, wodurch sich aktuelle Anwendungen sehr ähneln. Da die alten Konzepte aber, wie aus den wenigen Benutzertests ersichtlich, meist hinter den gewohnten 2D-Anwendungen zurückbleiben, ist dieser Weg eine Sackgasse.

Das Thema 3D wird außerhalb der Spielentwicklung also sehr stiefmütterlich behandelt. Es gibt nur wenige ernsthafte Projekte und diese sind meist nur vom Entwickler selbst dokumentiert, was Vergleiche und Forschung erschwert.

Das Hauptproblem bei den Nachforschungen zu diesem Artikel war die Einschränkung auf 3D-Dateiverwaltungssysteme, eines Teilgebeites innerhalb eines ohnehin schon schlecht dokumentierten Forschungsbereichs. Eine Erweiterung hätte aber zu Überschneidungen mit 3D-Desktops geführt, die in einer anderen Arbeit der Lehrveranstaltung betrachtet werden.

Quellen

1. 3D Graphics in the User Interface of a File System Browser, Niklas Höglund, Universität Stockholm, 2004
2. StepTree - A File System Visualizer, Thomas Bladh, MCS-2002-3, pp. 29, Dept. of Software Engineering and Computer Science, 2002
3. An Evaluation of Cone Trees, Andy Cockburn, Bruce McKenzie, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 2000.
4. 3D or not 3D? Evaluating the Effect of the Third Dimension in a Document Management System, Andy Cockburn, Bruce McKenzie, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 2001
5. Revisiting 2D vs 3D Implications on Spatial Memory, Andy Cockburn, Bruce McKenzie, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 2004
6. On a Web Browsing Support System with 3D Visualization, Toshihiro Yamaguchi, Hiromitsu Hattori, Takayuki Ito, and Toramatsu Shintani, Dept. of Computer Science and Engineering, Nagoya Institute of Technology, Nagoya, Japan, 2004
7. Elements of a Three-dimensional Graphical User Interface, Georff Lech, Ghassan Al-Quaimari, Mark Grieve, Noel Jinks and Cameron McKai, Department of Computer Science, Royal Melbourne Institute of Technology Melbourne, Australia, 1997
8. Cone Trees: Animated 3D Visualizations of Hierarchical Information, George G. Robertson, Jock D. Mackinlay and Stuart K. Card, Xerox Palo Alto Research Center, 1991
9. Extending Tree-Maps to Three Dimensions: A Comparative Study, Thomas Bladh, David A. Carr, and Jeremiah Scholl, Department of Computer Science and Electrical Engineering, University of Technology, Sweden, 2004

HOMEPAGES DER BEISPIELPROGRAMME

10. TDFSB: http://www.determinate.net/webdata/seg/tdfsb.html
11. TACTILE3D: http://www.tactile3d.com/
12. Xcruiser: http://xcruiser.sourceforge.net/
13. Clara: http://www.spatialknowledge.com/projects/clara/
14. SphereXPlorer: http://www.spheresite.com
15. StepTree: http://www.tibsoft.com/index.php?page=steptree
16. LG3D: https://lg3d.dev.java.net/
17. SpaceTime: http://www.spacetime.com/
18. B3: http://3b.net/browser/newhome.html

Die Internet-Quellen wurden alle zuletzt am 09.06.2007 besucht.

Fenstermanipulation in 2D Desktop-Umgebungen und Anwendbarkeit dieser Techniken an Touch Screens

Christian Kruschitz
Institut für Informatik Systeme
Universität Klagenfurt
christian.kruschitz@edu.uni-klu.ac.at

ABSTRACT
Fenster am Desktop zu manipulieren ist eine grundlegende Tätigkeit seit der Erfindung der Fenstermanager. Seit den 80er Jahren wurden viele Techniken gefunden um mit überlappenden als auch nicht-überlappenden Fenstern effizient arbeiten zu können. Diese Arbeit untersucht die wichtigsten Techniken der letzten Jahre und diskutiert sie anhand ihrer Funktion und Tauglichkeit. Abschließend werden kurz Szenarien vorgestellt wie Fenstermanipulationstechniken die mit den heute zur Verfügung stehenden Pointing-Methoden von Touch-Screens sinnvoll kombiniert werden können. Unter anderem wird auch das System von Microsoft vorgestellt: der berührungssensitve Wohnzimmertisch.

Author Keywords
Fenstermanipulation, Touch-Screens, Multitouch Interaktion

EINLEITUNG
Fenster-Manipulationstechniken sind innerhalb von Desktop Umgebungen die kritischsten Methoden um gute Performance bei der Erledigung von Aufgaben am Computer zu gewährleisten. Darum ist es notwendig effiziente Techniken zu entwickeln und diese in den heute benutzten Fenstermanagern zu implementieren. In den letzten Jahren gab es eine Reihe von neuen Ansätzen [7, 14, 15, 19–21] um die Interaktion mit Fenstern zu verbessern und zu erleichtern.

Diese Techniken funktionieren ganz gut auf herkömmlichen Displays die mit einer Mouse gesteuert werden. Aber wie ist es mit Touch Screens? Die Möglichkeit hier mit Fenstern zu interagieren geschieht nur über die Finger, die jedoch aufgrund ihrer Dicke nicht sehr präzise sind. Aber auch hier haben sich Wissenschafter einige Techniken [6, 12] einfallen lassen um das Auswählen von Objekten so präzise als möglich zu machen.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CHI 2006, April 22-27, 2006, Montréal, Québec, Canada.
Copyright 2006 ACM 1-59593-178-3/06/0004...$5.00.

Die oben genannten Techniken sind bei weitem nicht alle Methoden die entwickelt wurden. Im Rahmen dieser Arbeit möchte ich einige vorstellen, die meiner Meinung nach das State-of-the-Art der letzten 7 Jahre repräsentieren.

Diese Arbeit setzt sich zum Ziel, die oben genannten Techniken vorzustellen und diese in einem abschließenden Kapitel an Touch-Screens anzuwenden. Wobei diese Anwendbarkeit nur theoretischen Charakter, und dadurch nicht so sehr ins Detail geht, besitzt. Um genauere Erkenntnisse aus diesen Kombinationen zu gewinnen ist eine Implementierung in einem Prototypen notwendig. Ein weiterer Schritt wäre eine Evaluation der User-Experience dieser Systeme.

Das erste Kapitel widmet sich den Fenster - Manipulationstechniken. Vorgestellt und diskutiert werden drei *aktive Interaktionstechniken* und eine *passive Interaktionstechnik*.

Das zweite Kapitel konzentriert sich auf die Touch Screen Interaktionstechniken. Im Speziellen auf die Pointing- und Auswahlmethoden.

Das letzte Kapitel versucht die zuvor erklärten und diskutierten Methoden zu kombinieren und deren Anwendbarkeit aufzuzeigen. Abschließend wird ein System vorgestellt, dass diese Techniken bereits einsetzt.

FENSTER MANIPULATION
In den letzten Jahren bzw. Jahrzehnten sind die Fenstermanager bzw. die Technik wie man Fenster in Desktop-Umgebungen manipulieren kann, um schneller zum Ziel zu kommen, gleich geblieben. Jedoch ist die Anzahl der simultan zu bearbeiteten Tasks enorm gestiegen.
Kommerzielle Produkte wie die Betriebssystemfamilie von Microsoft Windows hat in den letzten Jahren kaum Neuerungen im Bezug auf Fenster-Manipulation mit sich gebracht. Die Forschung jedoch entwickelt immer wieder neue und recht erstaunliche Techniken um noch effizienter mit dem Computer arbeiten zu können. Vor allem bezieht sich das auf die Manipulation von Fenstern in Desktop - Umgebungen da heutzutage alle Betriebssysteme einen Fenstermanager implementiert haben und die Arbeit mit dem Computer nur über Fenster am Desktop führt. Deshalb ist es notwendig

effiziente Techniken zu finden bzw. zu entwickeln die dem User helfen seine Aufgaben schneller zu erledigen.

In dieser Arbeit teile ich die besprochenen Fenster - Manipulationstechniken in 2D-Desktop-Umgebungen in zwei unterschiedliche Bereiche ein. Zum einen ist das die *aktive Interaktion* mit Fenstern und zum anderen die *passive Interaktion* mit Fenstern.

Der erste Bereich definiert sich dadurch, dass die Interaktion mit einem Fenster auf Wunsch des Users zustandekommt. D.h. der User interagiert direkt mit dem Fenster. Um die Fenster direkt manipulieren zu können sind technische Hilfsmittel wie eine Mouse oder, wie wir später noch sehen werden, ein Touch Screen oder ähnliche Geräte notwendig. Hier möchte ich Beispielhaft die *Fold'n Drop* - Technik von Dragicevic [15] erwähnen. Bei dieser Technik geht es darum, dass der User mit Hilfe dieser Technik überlappende Fenster so manipulieren kann, dass er auch Zugriff auf nicht sichtbare Fenster hat, ohne einer Vielzahl an vorangegangenen Tasks ausführen zu müssen. Das Prinzip dahinter sind die sogenannten *Peeling Back Windows* [7]. Durch diese Technik werden die Fenster mit bestimmten Mouse-Gestiken umgebogen, um die darunterliegenden Fenster sichtbar zu machen. Eine Demonstration ist auf [16] zu sehen. Diese Technik wird in einem späteren Kapitel im Detail erklärt.

Der zweite Bereich zeichnet sich dadurch aus, dass der User nicht aktiv in das Geschehen eingreift. Dies sind die Bereiche in dem sich die Fenster durch vorgegebene Algorithmen ausrichten. Als Eingabe für die Algorithmen dienen die vom User manipulierten Fenster. Ein Beispiel dafür währe die *Elastic Window* Technik von Kandogan [22]. Das Prinzip dahinter ist, dass der freie Raum am Desktop durch Layout Algorithmen optimal ausgenutzt wird. Wenn der User ein Fenster größer macht, verkleinert sich die restlichen offenen Fenster proportional. Um ein Beispiel davon zu sehen, ist ein Video auf der Homepage der Autoren zu finden [23]. Des Weiteren basiert die Technik noch auf hierarchische Fenster und auf mehrfache Fenster-Operationen. Hierarchische Fenster-Operationen werden genutzt, um korrespondierende Fenster in einem Hauptfenster abzulegen. Ähnlich einem Papierstapel auf einem Schreibtisch. Da die *Elastic Windows* Technik nur indirekt mit dieser Seminararbeit zu tun hat, möchte ich auf [22] verweisen um einen detaillierteren Einblick in das Prinzip zu bekommen.

Aktive Manipulation

Die aktive Interaktion beschreibt die direkte Manipulation des Fensters durch den User. Zum Beispiel einen bestimmten Bereich eines Fenster ausschneiden. Im Folgenden werden drei *aktive Interaktionstechniken* beschrieben und diskutiert. Zu diesen drei zählen die *Fold'n Drop*-Technik von Dragicevic [16] [15], *Shrinking Window Operations* von Hutchings et al. [19] und die, der *Shrinking Windows Operation* Methode ähnlichen, *WinCuts* von Tan et al. [24].

Figure 1. Wenn die Mouse das Fenster mit einem Objekt verlässt dann knickt die Ecke ein wo das Objekt das Fenster verlassen hat. Quelle: [15]

Es ist mir durchaus bewusst, dass dies nicht alle Techniken sind die es für Fenstermanipulation gibt, jedoch würde eine genaue Beschreibung aller Techniken der letzten 7 Jahre den Umfang sprengen. Deshalb habe ich drei Techniken ausgewählt die das State-of-the-Art dieser Forschungsrichtung repräsentieren.

Fold'n Drop

Diese Methode versucht das Problem von *Dragging und Dropping* von Objekten zwischen überlappenden Fenstern zu verbessern. Um ein Objekt von einem Fenster zu einem anderen Fenster, das versteckt unter einem anderen Fenster liegt, zu verschieben, muss der User das Zielfenster erst einmal auffinden, um darin etwas ablegen zu können. In den heute implementierten Fenstermanagern setzt das eine Vielzahl von Aktionen voraus. Der User muss erst einmal das Zielfenster durch verschieben, verkleinern, minimieren oder sogar schließen von anderen Fenstern auffinden. Ist das Zielfenster erst einmal aufgefunden kann der User die eigentliche Aktion ausführen und sein Objekt vom aktuellen Fenster in das Vorgesehene zu verschieben. Um diese Aktionen zu minimieren wurde das Konzept der *Fold'n Drop* Methode entwickelt.

Die Methode basiert auf die in der Arbeit von Beaudouin-Lafon vorgestellten Technik der *Peeling Back Windows* [7] vereint mit der Methode der *Crossing-Based Interaction* von Accot [5].

Das Objekt wird im herkömmlichen Sinn geschoben und abgelegt, wobei hier, solange der Mouse-Button gedrückt ist, noch verschiedene *Folding* Interaktionen möglich sind. Im speziellen sind das :

- **Fenster verlassen:** Wenn der User ein Objekt aufgenommen hat und damit das Fenster verlässt, knickt an der Stelle, an der das Objekt das Fenster verlässt das Fenster ein und signalisiert dem User, dass es nun faltbar ist. Siehe Abbildung 1.
- **Bestätigen und falten der Fenster:** Nachdem man das Fenster mit dem Objekt verlassen hat ist es nun möglich das Falten zu bestätigen indem man das Fenster mit einer Bewegung zum Fenster faltet. Nun ist das Falten bestätigt und es ist möglich das Fenster jederzeit weiter zu falten. Siehe Abbildung 2.

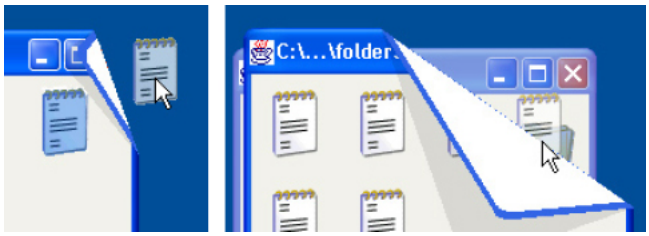


Figure 2. Das Fenster wird mit einer Mouse-Gestik zum Fenster hin gefaltet. Quelle: [15]

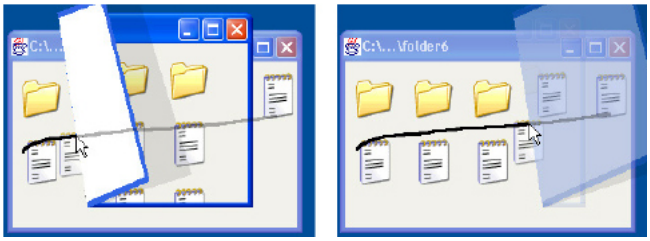


Figure 3. Durch eine lange Mouse-Gestik wird das Fenster verworfen. Quelle: [15]

- **Verwerfen von Fenster:** Es ist auch möglich ein Fenster zu verwerfen indem man es ganz faltet. Daraufhin verschwindet es vom Desktop. Siehe Abbildung 3.
- **Auffalten:** Um das Falten rückgängig zu machen, muss man mit der Mouse hinter die Falte und kann damit wie bei einem Blatt Papier von innen nach außen das Fenster glattstreichen, wie in Abbildung 4 zu sehen ist.
- **Mehrere Fenster falten:** Natürlich ist es auch möglich mehrere Fenster zugleich zu falten indem man die darüber liegenden Fenster die schon gefaltet sind einfach mit faltet.
- **Zurück zum Ausgangspunkt:** Um zum Ausgangspunkt zurückzukehren kann man den Vorgang abbrechen indem man mit der rechten Mouse-Taste klickt oder wenn der Vorgang abgeschlossen ist und das zu verschiebende Objekt an dem gewünschten Platz ist, werden die Fenster wieder in ihren Ausgangszustand versetzt.

Die Zeitersparnis bei der Ausführung von *Drag and Drop* Interaktionen innerhalb von verdeckten Fenstern mit Hilfe von *Fold'n Drop* ist offensichtlich. Dadurch, dass man sich das umständliche verschieben, verkleinern, schließen und minimieren erspart, und gleich zur Hauptaufgabe, dem Verschieben von Objekten kommt, spart man sich viel Zeit. Interessant wäre es, diese Technik in einem User-Test zu evaluieren, um zu sehen wie viel schneller bzw. effizienter User mit dieser Technik ihre Aufgaben am Computer lösen können. Solch eine Evaluation wäre sicher ein Argument um solche Techniken auch in kommerzielle Systeme zu implementieren.

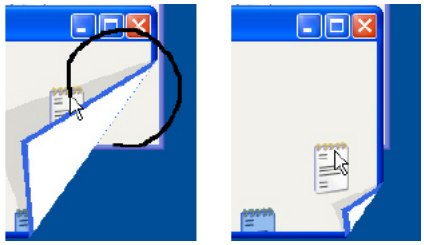


Figure 4. Von innen nach außen wird das Fenster glattgestrichen. Quelle: [15]

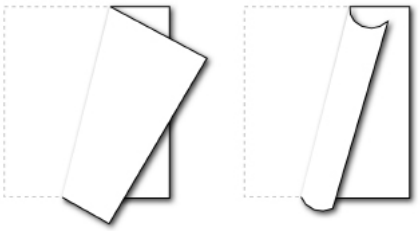


Figure 5. Falten(links) und Rollen(rechts). Dadurch bleiben Informationen des aufgerollten Fenster länger sichtbar. Quelle: [14]

Um einen Einblick in die Implementierung mittels dem Java Swing Framework zu bekommen sei auf [15] zu verweisen. Ein Demo-Video und eine Implementierung der Methode in Java ist auf der Homepage des Autors [16] zu finden. Weiters ist es auch möglich, was eher selten der Fall ist, diese Technik in der realen Anwendungsumgebung von Windows XP zu testen. Ein Programm, genannt *OriMado* [4], dessen Autor leider nicht herauszufinden war, implementiert die oben genannten *Folding* Interaktionen in Windows XP mittels VC.Net.

Des Weiteren wendet Chapuis et al. [14] diese Methode an um das *copy-and-paste* zwischen überlappenden Fenster effizienter zu machen. Er verändert jedoch die Falttechnik indem er die Fenster aufrollt. Dadurch wird das aufgerollte Fenster nicht so stark verdeckt und es bleiben die Informationen länger sichtbar. Abbildung 5 zeigt den Unterschied zwischen falten und rollen.

Shrinking Window

Shrinking Window Operations ist ein Ansatz um überflüssige Informationen auszublenden bzw. die relevanten Informationen eines Fensters auszuschneiden. Hutchings et al. stellt in seinem Artikel [19] eine Technik vor um das Platzmanagement am Desktop effizienter zu gestalten. Heutzutage ist es notwendig, aufgrund der Informationsbeschaffung, mehrere Fenster geöffnet zu haben um an Informationen zu kommen die zur Erfüllung einer Aufgabe notwendig sind. Jedoch sind diese Informationen, bedingt durch das Fensterdesign, mit unrelevanten Informationen angereichert.

Ein Fenster ist nach [19] in zwei Bereiche aufgeteilt. Der erste Bereich sind die Informationen die den User in erster Linie interessieren. Der zweite Bereich sind die Interaktionskomponenten um die Informationen innerhalb eines Fensters manipulieren zu können. Diese Komponenten werden

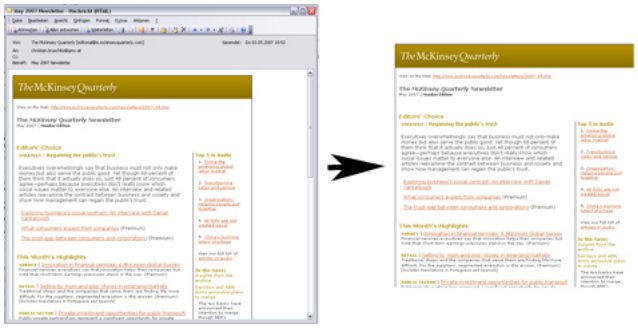


Figure 6. Ein Fenster das auf seine relevanten Informationen verkleinert wurde.

seltener gebraucht als die eigentlichen Informationen. Dadurch geht viel Platz am Desktop verloren, da die Komponenten einen beträchtlichen Teil eines Fensters in Anspruch nehmen. Vor allem, wenn ein Fenster inaktiv ist und nur als Informationsquelle dient, benötigen die Interaktionskomponenten viel wertvollen Platz.
Abbildung 6 zeigt ein Email-Fenster das auf die wesentlichen Informationen reduziert wurde. Hier sieht man das der verbrauchte Platz des Fensters um 20% minimiert wurde und dadurch Platz geschaffen wird um weitere Fenster am Desktop zu platzieren. Üblicherweise sind das Fensterleisten, Statuszeilen, Menüs usw. die bei dieser Operation "ausgeblendet" werden.

Um eben diesen Platz zu schaffen hat Hutchings et al. die Methode der *Shrinking Window Operations* entwickelt. Hierbei hat der User die Möglichkeit relevante Teile des Fensters auszuschneiden und damit den restlichen Teil des Fensters auszublenden. Dadurch ergibt sich eine Platzersparnis am Desktop. Diese Teile der Fenster kann der User dann beliebig am Desktop arrangieren. Würde man versuchen das Fenster auf die herkömmliche Weise zu verkleinern würden die Interaktionskomponenten mit verkleinert und nicht wie es beim Ausschneiden der Fall ist, ganz entfernt.

Diese Methode stellt dem User eine *snip*- und eine *unsnip*-Funktion zur Verfügung. Nachdem der User eine Region im Fenster bestimmt hat[1], ist es ihm möglich nur diesen Ausschnitt am Desktop anzeigen zu lassen. Dies geschieht zum Einen mit der *snip*-Funktion, welche das Fenster verkleinert und mit der *unsnip*-Funktion mit welcher das original Fenster wieder hergestellt werden kann. Angedeutet sind diese Funktionen in der ausgewählten Region durch einen Button bzw. ein Icon.
Ein *snipped*-Fenster kann vom User wie ein herkömmliches Fenster am Desktop platziert werden, jedoch ist es nicht möglich in diesem Modus das Fenster in herkömmlicher Art und Weise zu vergrößern oder zu verkleinern. Um das machen zu können, muss das Fenster zuerst mit der *unsnip*-Funktion auf die Originalgröße gebracht werden.
Abbildung 7 zeigt einen Desktop der ausschließlich mit *snipped*-Fenster arrangiert wurde. Hier ist gut zu sehen wie viel Platz diese Methode sparen kann.
Dem User ist es möglich mit einer globalen *snip*-Funktion alle Fenster auf einmal auf die gewählten Regionen schrumpfen zu lassen oder jedes einzelne Fenster individuell zu verkleinern.

Wie auch bei der *Fold'n Drop* Technik ist hier auch keine Evaluation der Methode erfolgt. Jedoch sieht der Autor die Probleme darin das die Evaluation schwierig ist, da man nicht den freigewordenen Platz am Desktop evaluieren kann, sondern wie effizient die Fenster arrangiert werden. Außerdem ist es nicht sinnvoll punktuell diese Technik zu evaluieren, da man eher auf das langfristige Verhalten bzw. wie diese Technik das Verhalten der User, auf lange Zeit gesehen, beeinflusst.
Meiner Meinung nach kann es dennoch zu Problemen führen, diese Technik anzuwenden. Vor allem wenn man einen Teil eines Dokumentes ausschneidet und ihn als Informationsquelle nutzt. Es ist selten das man nur einen Teil eines Dokumentes benötigt, sondern öfter das ganze Dokument. Um den nächsten Ausschnitt des Dokumentes zu erhalten ist es notwendig das Fenster mit der *unsnip*-Funktion wieder auf die Originalgröße zu bringen und dann den weiteren Teil des Dokumentes zu *snippen*. Dies führt wiederum zu Zeitverzögerungen beim Arbeiten.

Die nächste Methode, die im Rahmen dieser Arbeit vorgestellt wird, ist eine Technik mit sehr ähnlichem Konzept: Das Ausschneiden von relevanten Informationen. Jedoch ist es hier möglich im bereits verkleinerten Fenster zu scrollen. Dadurch fällt das ständige *snip* und *unsnip* weg und es ist möglich effizienter zu Arbeiten.

WinCuts

Ähnlich den *Shrinking Window Operations*, ist diese Methode entwickelt worden um Informationen am Desktop besser anordnen zu können. Denn die räumliche Anordnung spielt eine tragende Rolle bei der effizienten Bearbeitung von Aufgaben am Computer. Tan et al. [24] entwickelte die *WinCuts* Interaktions Technik, um relevante Informationen am Desktop mit Platzersparnis anzeigen zu können. Der Unterschied zur oben genannten Methode ist, dass die Fensterausschnitte nicht statisch sind, sondern live. D.h. der Ausschnitt zeigt immer aktuelle Daten an. Vor allem bei Daten respektive Informationen, die sich sehr schnell ändern, wie z.B: Börsenkurse. Des Weiteren ist es möglich mehrere Ausschnitte eines Fensters zu machen und diese Ausschnitte mit anderen Usern zu teilen.

Um ein neues WinCut zu erstellen muss der User einen Hotkey gedrückt halten und eine Region die ihn interessiert ausschneiden. Es können beliebig viele WinCuts gemacht werden. Entweder im gleichen Fenster oder in verschiedenen. Das WinCut hebt sich mit einer grünen gepunkteten Linie hervor. Das Fenster selbst bleibt unberührt und verhält sich

[1] aufziehen eines Rechtecks mit Hilfe der Mouse und gedrückt halten eines Buttons auf der Tastatur

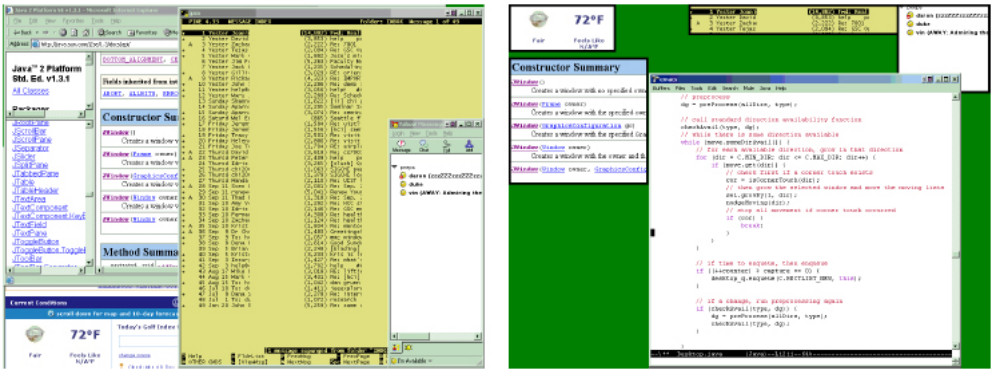


Figure 7. Links ein Desktop mit herkömmlichen Fenster. Rechts der gleiche Desktop mit Shrinking Window Operation. Quelle: [19]

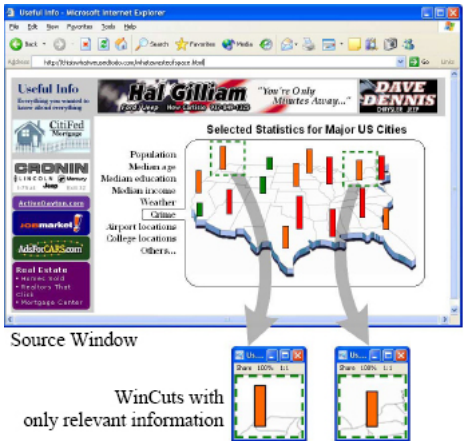


Figure 8. Zwei WinCuts eines Fensters um statistische Daten zu vergleichen. Quelle: [24]

Figure 9. Besprechung von Informationen durch remote WinCuts Quelle: [24]

wie ein herkömmliches Fenster(siehe Abbildung 8).

WinCuts enthält, wie bereits oben kurz erwähnt, live Repräsentationen des Inhalts. D.h. der User kann mit dem Inhalt in einem WinCut interagieren und die Daten manipulieren. Ein weiteres Feature ist, dass WinCuts mit anderen User ausgetauscht werden können. Jedoch sind die remote WinCuts nur read-only. Dies wird durch einen roten Rahmen um das WinCut angedeutet. Dadurch lassen sich Informationen von verschiedenen Usern gleichzeitig ansehen (siehe Abbildung 9).

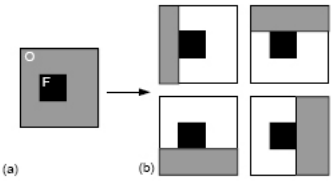


Figure 10. (a) zeigt ein schwarzes full-space Rechteck in einer leeren Szene, repräsentiert durch ein empty-space Rechteck (b) erzeugt 4 neue largest-empty-space Rechtecke.: [9]

Neben der räumlichen Anordnung von Informationen und deren Platzersparnis, entwickelte sich noch ein Werkzeug aus den WinCuts. Durch diese WinCuts ist ein *Rapid Prototyping Tool* entstanden. Durch ausschneiden verschiedener Teile einer Anwendung und neu arrangieren der Teile, kann schnell erkannt werden welches Interface am Besten funktioniert.

Passive Manipulation

Passive Manipulation ist die Manipulation von Fenstern die nicht direkt vom User gesteuert wird, aber durch seine Interaktion mit Fenstern beeinflusst wird.

Dynamic Space Management

Bell et al. stellt in seinem Paper [9] ein Konzept vor das den Desktop eines Users besser managed. Durch diverse Algorithmen werden Fenster am Desktop optimal ausgerichtet.
Genutzt werden sog. *full-space* Rechtecke, die die Dimensionen eines Objektes repräsentieren. Ein Objekt ist ein Fenster am Desktop. Die zweite Art von Rechtecken sind die *empty-space* Rechtecke. Diese repräsentieren den freien Platz am Desktop. Diese *emtpy-space* Rechtecke sind so groß, dass sie keine *full-space* Rechtecke überlappen. In Abbildung 10 sind alle Möglichkeiten eines *largest-empty-spaces* Rechtecks aufgezeigt. Es wird der größtmögliche Raum am Desktop alloziert. Dieser Space Manager bietet effiziente Algorithmen für inkrementelles hinzufügen und entfernen von *full-space* Rechtecke.
Beim hinzufügen von Objekten ist das Grundprinzip das der Reduktion[2]. Für eine genaue Erklärung dieses Prinzips ist

[2]die Erstellung von neuen kleineren *empty-space* Rechtecken aus einem existierenden *empty-space* Rechteck

der Artikel von Bernhard et al. [13] als weiterführende Literatur zu empfehlen.
Angewendet wird diese Technik wenn der User ein Fenster über stationäre Fenster legt, und damit die Sicht auf diese Fenstern versperrt. Hier greift der Algorithmus ein und schiebt das soeben vom User verschobene Fenster zum nächstliegenden freien Platz ohne diese Fenster zu überlappen. Für eine Präsentation dieser Technik ist ein Video auf der Homepage des Autor [10] zu finden.

Diese Technik findet unter anderem Anwendung in Augmented Reality [8]. Des Weiteren sind Anwendungen bei Spielen, Werbung, online Dokumente und e-Books angedacht [11] und überall wo es darum geht den verfügbaren Raum bestmöglich zu nutzen, ohne dass es zu Überlappungen kommt.

TOUCH SCREEN INTERAKTION

Touch Screens sind heutzutage sehr populär. Angefangen von kleinen PDAs bis hin zu Tablet PCs und öffentlichen Kiosk Systemen. Im Folgenden sind nach [25] die Vor- und Nachteile aufgezählt.

Vorteile:

- Direktes zeigen auf Objekte.
- Schneller als herkömmliche Pointing-Geräte.
- Die Finger sowie jede Art von Stift funktioniert auf der Oberfläche eines Touch Screens.
- Es ist kein Keyboard notwendig.
- Geeignet für Personen die nicht so oft mit dem Computer arbeiten, für Anwendungen die einen sehr häufigen Input benötigen und für Informationsanwendungen(z.B: Tourismusinformation).
- Intuitiv leicht erlernbar.

Nachteile:

- Es ist schwierig auf Ziele zu zeigen die kleiner als der Finger sind [6] (geringe Genauigkeit).
- Handbewegungen wenn der Touch Screen mit einem Keyboard genutzt wird. Der User muss seine Hand vom Keyboard wegbewegen um eine Interaktion am Bildschirm herbeizuführen.
- Müdigkeit der Arme (vertikal oder horizontale Ausrichtung des Displays)
- Schmutz an der Oberfläche beeinträchtigt die Sicht auf das Display.
- Abdeckung des Bildschirms durch die Hand des Users.
- Direkte Aktivierung der ausgewählten Funktion.

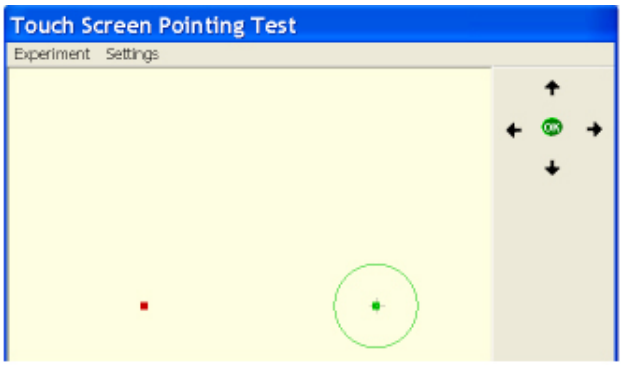


Figure 11. Mittels der virtuellen Pfeile kann das Crosshair am Objekt platziert und mittels OK-Button bestätigt werden. Quelle: [6]

Es wurde schon viel im Bereich der Touch Screens entwickelt. Vor allem die Interaktionstechniken, im speziellen im Bereich der Pointing-Methoden. In diesem Kapitel sollen dem Leser zwei Arten von Interaktionen mit Touch Screens, welche in den letzten Jahre entwickelt wurden, vorgestellt werden.

Zoom-Pointing

Zooming ist sicher eine der wichtigsten Interaktionstechniken an Touch Screens. Aufgrund der teilweise sehr kleinen Objekte ist es dem User nicht oder nur sehr schwer möglich mit dem Finger ein Objekt präzise auszuwählen. Aus diesem Grund wurde das *Zoom-Pointing* als Touch Screen Interaktionstechnik implementiert. Bei der Methode von Albinsson et al. [6] kann der User ein bestimmtes Objekt auswählen und vergrößern. Dies geschieht mittels eines Button. Ist dieser aktiviert ist es dem User möglich ein Rechteck mit dem Finger um das gewünschte Objekt zu zeichnen welches daraufhin vergrößert wird. Ein weiterer Button steht zur Verfügung um wieder auf die originale Größe zurückzukommen.

Virtual Keys

Eine weitere Technik von Albinsson et al. [6] sind die *Virtual Keys*. Diese Technik nutzt virtuelle Keys um einen Crosshair Cursor exakt auf einem gewünschten Objekt zu platzieren. Damit hat der Autor eine Technik geschaffen die auch ohne Zooming auskommt, um kleine Objekte auswählen zu können.
Der User platziert als erstes den Crosshair Cursor in der Nähe des gewünschten Objektes. Danach kann er den Cursor feinjustieren indem er Pfeile verwendet. Ist der Cursor über dem Objekt, bestätigt der User mittels OK seine Auswahl(siehe Abbildung 11).
Diese Technik ist im allgemeinen ein guter Ansatz um einen Cursor exakt auf einem Objekt zu platzieren, jedoch ergiebt sich dadurch auch eine Schwierigkeit. Es ist umständlich, da man den primären Finger vom Objekt wegbewegen muss um den Cursor durch die Pfeiltasten platzieren zu können. Dadurch geht der Vorteil der direkten Manipulation verloren.
Um weitere Techniken des Authors, wie *Cross-Keys*, *2D Lever* und *Precision-Handle* kennenzulernen sei der interessierte Leser auf [6] verwiesen.

Multi-Touch Pointing Techniques

Um *Zoom-Pointing* und *Virtual Key* anwenden zu können, genügt ein Finger. Damit die Effizienz gesteigert werden kann werden nun 2 Finger auf sogenannten Multi-Touch Screens angewandt. Benko et al. [12] entwickelte ein Konzept mit dem man mit seinem ersten Finger (pointing Finger) die Selektion vornimmt und mit dem zweiten Finger (non-pointing Finger) mittels einem Menüs den Modus des pointing Finger beeinflussen kann. Dadurch wird erreicht, dass der pointing Finger bei seiner Aufgabe nicht unterbrochen wird.
Im Folgenden werden die Techniken für multi-touch Pointing kurz präsentieren. Diese Techniken sind: *Dual Finger Offset*, *Dual Finger Midpoint*, *Dual Finger Stretch*, *Dual Finger X-Menu* und *Dual Finger Slider*.

Dual Finger Offset

Beim *Dual Finger Offset* ist der Cursor vom Finger um einen bestimmten Abstand versetzt. Um den Offset zu aktivieren muss man den non-pointing Finger irgendwie am Bildschirm platzieren. Um auch auf Links und Rechtshänder zu reagieren, ist der Cursor rechts bzw. links des pointing Finger platziert. Je nachdem wo der non-pointing Finger relativ zum primären Finger aufgesetzt wird.

Dual Finger Midpoint

Die *Dual Finger Midpoint* Methode vereint den Offset und die Möglichkeit die Cursorgeschwindigkeit zu steuern. Hierfür platziert man beide Finger auf dem Display und der Cursor wird genau in der Mitte des *primary* und *secondary* Finger gesetzt. Bewegt man jetzt beide Finger in die gleiche Richtung folgt der Cursor mit dieser Geschwindigkeit. Wird jedoch nur der pointing Finger bewegt folgt der Cursor mit der Hälfte der Geschwindigkeit des pointing Fingers. Klicken wird mit dem pointing Finger erreicht.
Probleme mit dieser Technik entstehen wenn die auszuwählenden Objekte kleiner als 2 Pixel sind oder wenn das Objekt in einer Ecke des Displays liegt.

Dual Finger Stretch

Eine weitere Methode ist die *Dual Finger Stretch* Technik. Diese Technik lehnt sich an *Zoom-Pointing* von Albinsson et al. [6] an. Der Vorteil der *Dual Finger Stretch* Methode ist es, dass die, wie bei *Zoom-Pointing* notwendigen Pfeile, nicht benötigt werden. Dadurch wird unterbrechungsfreies Arbeiten garantiert. Erreicht wird das indem der primary Finger auf die Region zeigt die vergrößert werden soll. Um diese Region bildet sich ein Rechteck welches dann mit dem secondary Finger aufgezogen und dadurch die Vergrößerung erreicht wird. Nimmt der User den non-pointing Finger vom Display, wird die Vergrößerung aufgehoben und der Cursor des pointing Fingers hat einen Offset. Dieser Offset bewirkt, dass der primary Finger auf das zuvor ausgewähltes Objekt zeigt ohne das der Finger das Objekt verdeckt.

Dual Finger X-Menu

Beim *Dual Finger X-Menu* ist der non-pointing Finger für die Auswahl des Modus verantwortlich. Aktiviert wird es dadurch, dass der non-pointing Finger das Display berührt

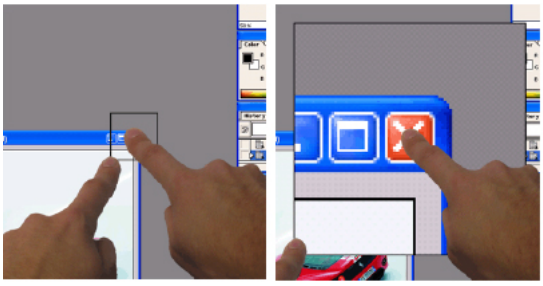


Figure 12. Dual Finger Stretch vergrößert mit Hilfe des secondary Finger einen Teil des Displays um präziser Auswählen zu können. Quelle: [12]

Figure 13. Ein Anwendungsbeispiel des Dual Finger X-Menu. Der User wählt mit dem non-pointing Finger den 10xslow Modus aus um den Cursor auf das Schließen Symbol des Fensters zu bekommen. Als Feedback für den User ist der Cursor mit zwei Kreisen umgeben was beduetet das er sich im 10x slow Modus befindet. Quelle: [12]

und ein Kreis Menü erscheint. Der Finger befindet sich in der Mitte des Menüs. Das Menü hat sechs Buttons zu Auswahl. Vier davon sind für die Geschwindigkeit des Cursors (*normal speed*, *slow 4x*, *slow 10x* und *freeze*). Die übrigen Zwei, *snap* und *magnify* dienen als Hilftools zur Kontrolle des Cursors und für das Zooming. Der *snap*-Modus entfernt den Offset vom Cursor und der *magnify*-Modus stellt eine Vergrößerung in der Mitte des Menüs dar. Vergrößert wird der Bereich unter dem Cursor. Abbildung 13 zeigt das Menü mit ausgewähltem *slow 10x*-Modus.

Dual Finger Slider

Eine weitere Technik ist der *Dual Finger Slider*. Bei dieser Technik muss der User seinen non-pointing Finger in Richtung des pointing Finger bewegen um die Geschwindigkeit Cursor – Pointing Finger einzustellen. Dies geschieht in drei Schritten. Die gleichen Abstufungen sind im Dual Finger X-Menu verfügbar. Diese Schritte sind: normal, slow4x, slow 10x und freeze. Um wieder normale Geschwindigkeit einzustellen, muss der User den non-pointing Finger vom pointing Finger wegbewegen.

ANWENDUNG VON FENSTERMANIPULATIONS - TECHNIKEN AN TOUCH SCREENS

Die letzten Kapitel dieser Arbeit beschäftigten sich mit den Techniken die meiner Meinung nach zur Zeit das State-of-the-Art der Fenstermanipulation und der Pointing Methoden an Touch Screens ist. Es ist auch einsichtig, dass nicht alle Techniken vorgestellt werden können die in den letzten 7 Jahren entwickelt und publiziert wurden.

In diesem Kapitel wird eine Diskussion begonnen inwieweit die Fenstermanipulationstechniken mit Touch Screens kompatibel sind bzw. ob es möglich ist diese Techniken auch an Touch Screens einzusetzen. Der limitierende Faktor beim Einsatz der Fenstermanipulation ist ohne Zweifel die Genauigkeit bei der Auswahl von Objekten. Es werden auch selten vollständige Fenstermanager mit Touch Screens genutzt da der Umgang mit Fenstern relativ schwierig ist aufgrund der beschränkten Möglichkeiten diese zu manipulieren. Deshalb werden diese Systeme oft mit Anwendungen ausgestattet, die es dem User ersparen, Fenster verschieben, verkleinern, usw. zu müssen. Diese Anwendungen füllen den ganzen Bildschirm aus und alle wichtigen Funktionen sind für den User sichtbar. Der Vorteil liegt darin, dass die Anwendungsfunktionen einfach angeordnet sind und der User dadurch das System sehr schnell erlernt. Jedoch ist die Funktion dieser Anwendungen sehr eingeschränkt und nur auf einen bestimmten Zweck ausgerichtet. Dem User ist es nicht möglich den Desktop anzupassen bzw. selbständig zu organisieren.

Wie würde die Technik von Dragicevic [15] kombiniert mit einem Touch Screen sich verhalten? Hierbei entstehen die ersten Probleme da der Finger nicht so präzise ist wie die Mouse. Um die Technik ausnutzen zu können, sollte man die Objekte zum Verschieben präzise auswählen können. Hier könnte die zooming Technik *Dual Finger Stretch* [12] zum Einsatz kommen. Mit dieser ist es möglich mit zwei Fingern zu arbeiten, was dem User meiner Meinung nach ein sicheres Gefühl beim Ausführen der Tätigkeit gibt. Wenn nun das Objekt, durch *Dual-Finger-Stretch*, ausgewählt ist, kann man die *folding*-Technik einsetzten indem man Gesten vollführt, um die Fenster zu falten. Probleme könnten hier entstehen durch den erforderlichen ständigen Kontakt des Fingers mit dem Display. Denn sobald der Kontakt zum Display unterbrochen wird, verliert man das Objekt welches in ein anderes Fenster verschoben werden sollte. Dadurch wiederum gehen alle Fenster wieder zum Ausgangspunkt zurück. Ob es sehr ermüdend auf den User wirkt wenn er diese Aufgabe ausführt, hängt auch davon ab wie der Touch Screen installiert ist. Bei einer horizontalen Installation ist die Ermüdung bei weitem nicht so groß wie bei einer vertikalen Installation des Bildschirms.
Die Technik von Dragicevic [15] und Benko [12] richtig eingesetzt, ist auf jeden Fall ein guter Ansatz um diese Fenstertechnik auch bei Touch Screens einsetzen zu können.

Um *Dynamic Space Management* bei Touch Screens einsetzten zu können, sind grundsätzlich keine zusätzlichen bzw. verbesserten Interaktionstechniken nötig. Diese Techniken arbeiten mit Algorithmen die auf die Platzierung von Fenstern reagieren. Es muss nur dem User möglich sein Fenster selbständig am Display platzieren zu können und der Fenstermanager muss eine Überlappung der Fenster zulassen.

WinCut in Touch Screens zu verwenden ist eine guter Ansatz um Fenster an Displays benutzerdefiniert, ohne überflüssige Informationen, anordnen zu können. Diese Technik verlangt eine präzise Auswahl der auszuschneidenden Region. Um das zu gewährleisten sind Techniken wie *Dual-Finger X-Menü* kombiniert mit dem *Dual-Finger Stretch* geeignet. Um eine Region auswählen zu können, und das relativ exakt, nutzt man die Funktionen zum verlangsamen des Cursors um genau eine bestimmte Region auswählen zu können. Um sie dann auszuwählen, nutzt man das Prinzip des *Dual-Finger Stretch*, indem man ein Rechteck, mit dem noch immer verlangsamten Cursor, aufzieht. Anstatt diesen Bereich dann zu vergrößern wird er bestätigt und mit einer weiteren Bestätigung durch einen Klick auf einen Button dann das betreffende Fenster ausgeschnitten. Diese Funktionen, wie der Button zum Bestätigen des Ausschneidens oder der Share-Button, um *WinCut*-Fenster mit anderen User zu teilen, können leicht in das *Dual-Finger X-Menü* eingebunden werden.

Microsoft Surface

Microsoft hat Anfang Juni 2007 ein Konzept vorgestellt das nach eigenen Angaben den Gebrauch des Computers revolutionieren soll. Genannt wird es *Microsoft Surface* [3]. Dieses System besteht aus einem 30 Zoll großen Bildschirm der in die Tischplatte integriert ist. Als Pointing Device werden die Finger der Akteure genutzt. Eine Besonderheit ist dass das Display zwischen 52 verschiedenen Fingern unterschieden kann. D.h. es ist ein multitouch-fähiges Display. Erkannt werden die Bewegungen der einzelnen Finger durch 5 infrarotempfindliche Kameras. Infrarot deshalb da das Gerät in der natürlichen Umgebung eines Raumes aufgestellt ist und das Restlicht die Unterscheidung der Finger stören würde. Das Bild wird von einem Projektor von unten auf die Tischplatte projeziert. Als Steuereinheit dient ein PC mit Windows Vista als Betriebssystem. Eine schematische Darstellung ist in Abbildung 14 zu finden.

Zur Zeit hält sich Microsoft noch bedeckt welche Technologien, im Speziellen die Pointing Methoden, eingesetzt wurden. Schaut man genauer hin erkennt man, dass Techniken eingesetzt wurden die schon im Vorfeld an diversen Prototypen von verschiedenen Forschungsgruppen vorgestellt wurden. Ein Beispiel dafür ist die Zooming Technik. Um ein Bild, in diesem Fall ein Foto, zu vergrößern nutzt man 2 Finger. Diese 2 Finger müssen das zoomende Objekt aktivieren und anschließend bewegen sich die 2 Finger voneinander weg und dadurch vergrößert sich das gesamte Objekt. Eine ähnliche Technik wurde schon von Benko et al. [12] vorgestellt. Die sogenannte *Dual Finger Stretch* Technik. Auch

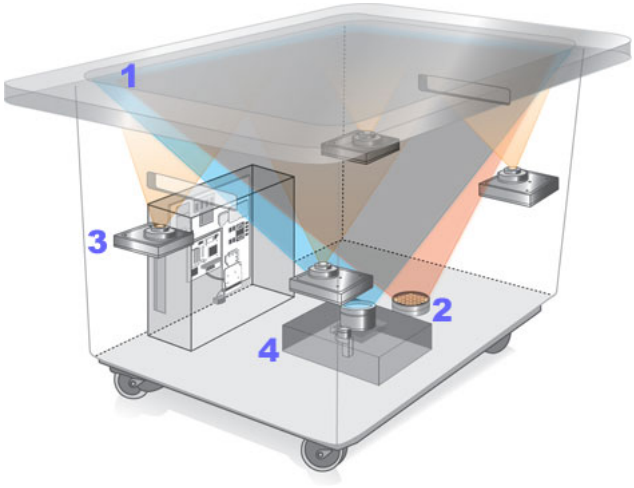


Figure 14. (1)Projektionsfläche (2)Infrarotquelle (3) PC (4) Projektor Quelle: Intoaroute

beim IPhone [1] von Apple kommt diese Methode zum Einsatz um Bilder zu vergrößern.

Ähnliche Systeme wie das von Microsoft wurden bereits vor Jahren vorgestellt. Jedoch verfügte keiner dieser Gruppen die notwendigen finanziellen Mittel um ihr Produkt so zu promoten.
Erwähnenswert sind die Arbeit von Han [2, 18]. Er entwickelte eine Multitouch Wall um direkt mit den Objekten interagieren zu können.

Multi-Touch Interaction Wall

Han entwickelte eine *Multi-Touch Interaction Wall* [2, 18] die 5 Meter lang und 1 Meter hoch ist (siehe Abbildung 15). Sie besitzt die Möglichkeit, ähnlich dem *Microsoft Surface*, mehrere Finger gleichzeitig erkennen zu können. Dadurch wird es möglich, dass mehrere Personen gemeinsam an dieser interaktiven Wand arbeiten können. Verstärkt wird dies dadurch dass die Wand horizontal ausgerichtet ist. Dadurch ist es für User leichter, gemeinsam die Wand zu benutzen. Aber aus der horizontalen Anbringung der Wand entsteht auch das Problem dass es sehr schnell zu Ermüdungserscheinungen in den Armen der User kommt.

Frustrated total internal reflection [17], ist die von Han entwickelte Technik um berührungsempfindliche große Flächen zu ermöglichen. Das Prinzip stammt aus den 60ern. Hier wurde sie bei Fingerabdruck-Bilder eingesetzt. Infrarotes Licht wird an den Ecken der Oberfläche eingespeist. Dies wird dann innerhalb der Oberflächenplatte reflektiert. Ist man in Berührung mit der Oberfläche, geht an dieser Stelle Licht verloren. Dadurch kann man feststellen wo man sich am Display mit dem Finger gerade befindet. Ein weiterer Vorteil dieser Technik ist es, dass sie beliebig skaliert werden kann. Nur deshalb können auch so große berührungsempfindliche Displays entwickelt werden.

Figure 15. Multi-Touch Wall Quelle: www.perceptivepixel.com

Die Applikationen die diese Wand bietet, sind intuitiv zu bedienen und die Bedienbarkeit ist schneller als mit herkömmlichen Pointing Geräten. Google Earth mit den Händen zu bedienen ist sehr viel effizienter. Mit einfachen Bewegungen der Finger, ähnlich dem *Dual Finger Stretch*, ist es möglich zu zoomen, kippen und rotieren der Bilder.
Aber um Alltagstauglich zu werden ist es sinnvoll Fenstermanager zu implementieren die in heutigen Desktop Umgebungen üblich sind.

KONKLUSION

Durch diese Arbeiten sieht man, dass es in der Zukunft sehr viel Potential für multitouch Systeme gibt. Aufgrund ihrer schnellen Bedienbarkeit sind sie den klassischen, mit der Maus zu bedienenden Point and Klick Oberflächen im Vorteil. Durch diese neuen Interaktionsarten eröffnen sich viele neue Interaktionsmöglichkeiten. Auch die Darstellung von Informationen wird dadurch beeinflusst.
Jedoch sind in diesen Systemen eigens dafür entwickelte Fenstertechniken implementiert worden. In Zukunft sollte es möglich sein, die aktuellen Fenstermanager auch auf diesen Oberflächen nutzen zu können. Dadurch wird der Umstieg vom derzeit üblichen Desktop PC zu einem multitouch System wesentlich vereinfacht.

DANKSAGUNG

Ich möchte meinen Kollegen für die konstruktiven Reviews danken.

REFERENCES

1. Apple IPhone. Website, Juni 2007. `http://www.apple.com/iphone/`.
2. Demo der Multitouch Wall von Jefferson Han. Website, Juni 2007. `http://www.perceptivepixel.com/`.
3. Microsoft Surface. Website, Juni 2007. `http://www.microsoft.com/surface/`.
4. OriMado: Implementierung und Source Code der “Fold’n Drop” Technik in Windows XP. Website, Mai 2007. `http://www.kmonos.net/lib/orimado.en.html`.

5. J. Accot and S. Zhai. More than dotting the i's — foundations for crossing-based interfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 73–80, New York, NY, USA, 2002. ACM Press. weiterführende Literatur.

6. P.-A. Albinsson and S. Zhai. High precision touch screen interaction. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 105–112, New York, NY, USA, 2003. ACM Press.

7. M. Beaudouin-Lafon. Novel interaction techniques for overlapping windows. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 153–154, New York, NY, USA, 2001. ACM Press.

8. B. Bell, S. Feiner, and T. Hollerer. View management for virtual and augmented reality. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 101–110, New York, NY, USA, 2001. ACM Press.

9. B. A. Bell and S. K. Feiner. Dynamic space management for user interfaces. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 239–248, New York, NY, USA, 2000. ACM Press.

10. B. A. Bell and S. K. Feiner. Demonstration video of dynamic space management. Website, Mai 2007. `http://www.cs.columbia.edu/~blaine/SpaceManager/final_384.wmv`.

11. B. A. Bell and S. K. Feiner. Dynamic space management for user interfaces. Website, Mai 2007. `http://www1.cs.columbia.edu/~blaine/SpaceManager/`.

12. H. Benko, A. D. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272, New York, NY, USA, 2006. ACM Press.

13. M. Bernard and F. Jacquenet. Free space modeling for placing rectangles without overlapping. *Journal of Universal Computer Science*, 3(6):703–720, 1997. `http://www.jucs.org/jucs_3_6/free_space_modeling_for`.

14. O. Chapuis and N. Roussel. Copy-and-paste between overlapping windows. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 201–210, New York, NY, USA, 2007. ACM Press.

15. P. Dragicevic. Combining crossing-based and paper-based interaction paradigms for dragging and dropping between overlapping windows. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 193–196, New York, NY, USA, 2004. ACM Press.

16. P. Dragicevic. Fold'n Drop. Website, Mai 2007. `http://www.dgp.toronto.edu/~dragice/foldndrop/`.

17. J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA, 2005. ACM Press.

18. J. Y. Han. Multi-touch interaction wall. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Emerging technologies*, page 25, New York, NY, USA, 2006. ACM Press.

19. D. R. Hutchings and J. Stasko. Shrinking window operations for expanding display space. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 350–353, New York, NY, USA, 2004. ACM Press.

20. E. W. Ishak and S. Feiner. Content-aware layout. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 2459–2464, New York, NY, USA, 2007. ACM Press.

21. E. W. Ishak and S. K. Feiner. Interacting with hidden content using content-aware free-space transparency. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 189–192, New York, NY, USA, 2004. ACM Press.

22. E. Kandogan and B. Shneiderman. Elastic windows: improved spatial layout and rapid multiple window operations. In *AVI '96: Proceedings of the workshop on Advanced visual interfaces*, pages 29–38, New York, NY, USA, 1996. ACM Press.

23. E. Kandogan and B. Shneiderman. Elastic Windows. Website, Mai 2007. `http://www.cs.umd.edu/hcil/elastic-windows/`.

24. D. S. Tan, B. Meyers, and M. Czerwinski. WinCuts: manipulating arbitrary window regions for more effective use of screen space. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1525–1528, New York, NY, USA, 2004. ACM Press.

25. G. Waloszek. Interaction design guide for touchscreen applications (experimental). Website, Mai 2007. `http://www.sapdesignguild.org/resources/TSDesignGL/`.

3D Desktop Effekte

Marlene Stroj
Universität Klagenfurt
9020 Klagenfurt, Österreich
marlene@edu.uni-klu.ac.at

ABSTRACT
Fast alle modernen Betriebssysteme bieten dem User ein Window System um den Platz auf ihrem Desktop zu verwalten. Programmfenster und Icons „kämpfen“ um einen Platz am Desktop, die Navigation wird bei vielen geöffneten Fenstern immer schwieriger. Zudem werden Displays vor allem auf mobilen Geräten immer kleiner, das Platzmanagement eine immer größere Herausforderung. Einige User greifen auf mehrere Monitore zurück, um das Platzproblem zu lösen. In den letzten Jahren wurden aber auch viele Softwarelösungen zu diesem Problem entwickelt. Neue Window Manager können wesentlich mehr als einfach nur mehrere Monitore zu simulieren. User werden mit Konzepten wie 3D-Desktops, transparenten Fenstern, Zooming Windows und ähnlichem konfrontiert. Es gibt eine Reihe von grafischen Möglichkeiten, um Fenster und Desktop-Items zu platzieren. Beispiele sind Task Gallery[7] von Microsoft, Metisse[3], 3D-Desktop[11], Yod’m 3D[15] und andere. Solche Effekte in User Interfaces bieten zwar viele Möglichkeiten, allerdings steigt die Komplexität des Interfaces um ein Vielfaches und die Navigation wird komplizierter. Diese Lösungen wurden entwickelt, um dem User das Platzmanagement zu erleichtern und ihm, trotz vieler offener Fenster, die schnelle Erfüllung seiner Tasks zu ermöglichen. Sind diese Effekte jedoch wirklich hilfreich, oder nur dazu da, den User zu beeindrucken? In diesem Paper sollen solche Desktop Effekte und konkrete Implementierungen von Window Managern, die mit diesen Effekten arbeiten, vorgestellt werden.

Autor Keywords
3D Desktops, 3D Effekte

EINLEITUNG
Wir leben in einer dreidimensionalen Welt und während diese Dreidimensionalität in diversen Computerspielen schon lange Einzug gehalten hat, ist der Desktop vieler Anwender noch immer 2D. Dabei gibt es eine ganze Reihe von Forschungsaktivitäten in diesem Bereich und auch in neuen Betriebssystemen, wie zum Beispiel Microsoft’s Windows Vista [25], kommt der eine oder andere 3D-Effekt ins Spiel. Aber inwieweit sind diese Effekte hilfreich? Können sie den User bei seiner Arbeit wirklich unterstützen? Die im Folgenden vorgestellten Projekte und Lösungen sollen einen Einblick in die laufenden Forschungsarbeiten der unterschiedlichen Softwarehersteller beziehungsweise der Open Source Community geben.

MICROSOFT
Bereits 1999 startete Microsoft mit Task Gallery, einem 3D Task Manager, die Forschung im Bereich der 3D-Desktops. 3D-Effekte wurden aber erst im neuen Windows Vista[25] in das Betriebssystem Windows aufgenommen und auch jetzt nur sehr spärlich eingesetzt.

Task Gallery: Eine Galerie für Tasks
Task Gallery [7] ist ein von Microsoft entwickelter Prototyp eines 3D Task[1] Managers. Task Gallery nutzt das räumliche Gedächtnis der User für das Task Management. Dabei wird die Metapher des Desktops durch die einer Kunstgalerie ersetzt. Die Programmfenster können wie Bilder links und rechts an den Wänden hängen oder an der Decke „kleben“ bzw. am Boden liegen. Um dem User die Orientierung zu erleichtern, wurde ein einfacher Gang als Darstellung für die virtuelle 3D Welt gewählt. Nach vorne hin ist die Galerie mit einer Wand abgeschlossen vor der eine Bühne steht. Zusätzlich wird eine Ansicht aus der Vogelperspektive angeboten, die alle in der Galerie vorhandenen Tasks zeigt.

Die Navigation ist beschränkt auf Vor- und Rückwärtsbewegungen wobei der User sich endlos rückwärts bewegen kann und dabei immer mehr Räume erscheinen. Die Aufteilung des 3D Raums in mehrere Räume soll dem User helfen, sich besser zurechtzufinden und sich an die Plätze der einzelnen Task Fenster zu erinnern, da ungeübte User vor allem in virtuellen 3D Welten leicht die Orientierung verlieren. [7]

Mit einem Klick auf einen Task wird er auf die Bühne am Ende der Galerie befördert und kann dort bearbeitet werden. Zuerst wird der zuvor auf der Bühne gewesene Task wird an seine alte Position in der Galerie zurückbewegt und dann der aktuell ausgewählte Task auf die Bühne bewegt. Ein „Geist“ des Tasks bleibt als Markierung der an der alten Position in der Galerie zurück.

[1] Task wird als Ansammlung einer Reihe von Dokumenten und Applikationen angesehen, die der Erfüllung einer bestimmten Aufgabe dienen. [7, S. 1]

Kurze Animationen der Aktionen werden als Übergang verwendet, damit der User diese nachvollziehen kann und die Orientierung nicht verliert. [7]

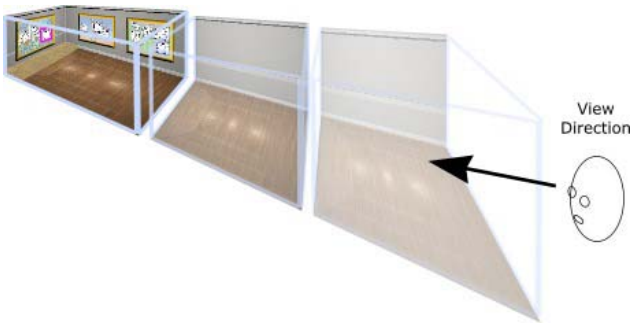


Abbildung 1. Task Gallery: Unterteilung des virtuellen Raums in mehrere Räume als Orientierungshilfe für den User [23]

Neu angelegte Tasks werden am Boden vor der Bühne abgelegt und können vom User selbst wie gewünscht in der Galerie platziert werden. Alle Tasks können beliebig zwischen den Wänden bzw. Decke und Boden der Galerie hin und her bewegt werden. Task Gallery funktioniert mit 2D Windows Applikationen, da die Software mit Redirection dieser arbeitet. Eines der Hauptprobleme der Implementierung war, beim Start alle für die Tasks benötigten Applikationen wieder zu öffnen, um das Layout wiederherzustellen, dass der User beim letzen Verlassen des Task Managers gesehen hat. [7]

Abbildung 2. Sicht auf die Bühne und Tasks an den Wänden und Decke der Task Gallery [7]

Usabilitytests haben ergeben, dass das Tool (mit einigen Verbesserungen des ersten Prototyps) User beim Management von Tasks gut unterstützt und die Verwendung auch einen gewissen Spaßfaktor für den User hat [7]. Der User wird mit visuellen und auditiven Hinweisen und Hilfestellungen (Animationen der ausgeführten Aktionen, Programmfenster werfen Schatten, etc.) bei der Orientierung und Navigation unterstützt. Die Galeriemetapher eignet sich, zusammen mit den audio-visuellen Hilfestellungen, gut für die Gestaltung des 3D Raumes, da eine Kunstgalerie jedem bekannt ist [9]. Probleme bereitete die schlechte Lesbarkeit der Texte in den Fenstern, die an den Wänden, dem Boden oder der Decke waren. Im Allgemeinen ergaben die Usabilitytests, dass die User sich bei der Interaktion mit dem Interface zufrieden zeigten, ihnen jedoch nicht immer sofort ganz klar war, was zu tun ist [7]

Das Projekt war als Research Prototyp gedacht, um Ideen und Erkenntnisse über Interaktion, Metaphern und den Umgang von Usern mit einer 3D Umgebung zu gewinnen. Die Arbeit an dem Projekt wurde nicht direkt fortgesetzt, allerdings flossen die Erkenntnisse, laut Microsoft, in andere Projekte mit ein [23], die unter [24] zu finden sind.

3D in Windows Vista

Windows bietet in seinem neuen Betriebssystem *Vista*[25] erstmals einen bereits „eingebauten“ 3D Effekt an, nämlich zum Switchen zwischen den offenen Programmfenstern. *Flip-3D* [26] nennt sich dieses Feature, das alle offenen Fenster inklusive dem Desktop selbst als Thumbnails hintereinander gestaffelt in einer 3D-Ansicht zeigt und dem User schon beim Switchen selbst ein Bild des Programmfensters zeigt. Das Feature wird allerdings zusätzlich zum normalen *Flip* (wie bisher mit den Tasten ALT+TAB) angeboten, welches beim Switchen einfach Thumbnails der offenen Fenster nebeneinander anzeigt ohne Spezialeffekt. [25] Daher die Frage, ist der Effekt sinnvoll oder nur ein Ressourcenverbrauchendes „eye candy“ um Mac-User abzuwerben. Viele sind der Meinung, dass Windows damit nur Apple’s Exposé imitieren wollte und das Augenmerk nicht wirklich auf einer verbesserten Usability für die User liegt. Wissenschaftliche Arbeiten über Usabilitytests wurden in diesem Zusammenhang nicht veröffentlicht [6], jedoch bieten „Selbstversuche“ eines Reporters und eines Bloggers einen Einblick [17, 18]. Subjektiv von diesen Testern gesehen, kann weder Windows Vista noch Flip3D überzeugen und ein (nicht gelungener) Versuch von Windows näher an die Vorgabe, die Mac OS X macht, heranzukommen.

APPLE: MAC OS X

Die grafische Gestaltung der Icons gibt der Oberfläche des Mac OS X ein wenig 3D Aussehen. Dennoch werden auch hier 3D Effekte spärlich eingesetzt. Das Aqua User Interface[12] bietet neben dem Fisheye-View[2] der Menüleiste auch eine 3D-Würfelanimation für das schnelle Switchen zwischen User Accounts. Diese Animation wird in anderen Window Manager häufig zum Switchen zwischen virtuellen Desktops verwendet [11, 14, 15, 16].

Ein weiteres Feature ist *Exposé* (Abbildung 3). Mit einfachem drücken von einer Taste können der Desktop

[2] Ein „Fisheye-View“ bezeichnet das Hervorheben von Details in der Mitte und der Verkleinerung des Umfeldes am Rande, als ob eine Linse über den Untergrund gelegt werden würde.

ausgezoomt und die Programmfenster arrangiert werden. Das Anklicken eines Fensters zoomt es heran und aktiviert es zur Bearbeitung. Dieser *Exposé-Effekt* wurde von mehreren Window Managern (siehe die Beryl[14], Compiz[16] und Metisse[3] Window Manager) kopiert beziehungsweise sehr ähnliche Effekte übernommen.

Abbildung 3. Der Exposé Effekt: Ein Übersicht über alle offenen Programmfenster [12]

Ein 3D Cube Dashboard

Am 3. Mai 2007 ließ Apple die so genannten *Multiple Dashboards* vom US Patent & Trademark Office patentieren. Damit ist eine Würfelansicht gemeint, auf dessen Oberflächen User Steuerungs-Widgets platzieren können.[3] Kontrollmechanismen wie der Scrollbar oder ein Tastendruck können manipuliert werden, um die Animation – also das Drehen des Würfels um eine seiner Achsen – zu steuern. [13] In der Implementierung ist die Form der Dashboards ist nicht auf einen Würfel beschränkt, sondern kann jede beliebige 3D Form (Zylinder, Diamant, Kugel, …) sein. Ob dieses Feature schon im neuen OS X Leopard enthalten sein wird, ist allerdings noch offen (siehe Anhang, Screenshot 1). [13]

SUN'S LG3D: THROUGH THE „LOOKING GLASS"

Suns Java 3D basiertes 3D Interface Projekt nennt sich *Looking Glass*[28] und bietet dem User wesentlich mehr 3D Features als Windows Vista. Das Projekt wurde 2004 von Hideya Kawahara ins Leben gerufen und ging später Open Source. Es wurde bei der Java One Konferenz 2004 vorgestellt.

Das primäre Ziel war nicht ein generisches 3D API zu entwickeln, sondern sich auf die 3D Erfahrung des Users zu konzentrieren und diese zu verbessern. Dabei stehen Produktivität des Users und der Spaß bei der Verwendung des Interfaces für den User im Vordergrund. Man setzte sich zum Ziel, vor allem die Usability des Interfaces durch den Einsatz von 3D Grafiken zu verbessern. Jedenfalls zumindest keinen Rückschritt hinter die Usability von 2D Interfaces zu machen, und sich nicht nur auf die 3D-Grafik selbst zu konzentrieren. Durch Looking Glass sollte nicht das 2D Interface verworfen und dem User stattdessen eine 3D Welt geboten werden, sondern eine schönere, mit 3D Elementen verbesserte Version des 2D Interfaces. [**Fehler! Verweisquelle konnte nicht gefunden werden.**] Input und Output erfolgt über die klassischen Standarddevices wie Mouse, Keyboard und Monitor. Das Ziel des Projektes sollte ein globales 3D Paradigma sein, die auf das User Interface und das Interagieren des Users mit dem Interface angewendet werden kann. [30]

Nach dem erfolgreichen proof-of-concept Test ließen sich folgende Erfolgsfaktoren von LG3D identifizieren:

- die gute Integration von bestehenden 2D Applikationen
- das Erleben des Interfaces als gelungene Erweiterung des gewöhnlichen 2D Interfaces durch 3D Elemente; Eine 2½D Sicht, Ähnlichkeiten zu den bestehenden 2D Interfaces erhaltet
- besonders gutes Feedback des Systems für den User und
- die besondere Beachtung der visuellen Repräsentation.

Darauf aufbauend wurden Designkriterien entwickelt, nach denen Looking Glass weiter entwickelt werden wird.

Abbildung 4: Looking Glass 3D-Ansicht des CD Players [28]

In Zukunft soll Looking Glass stabiler und die Usability des Interfaces noch gesteigert werden. Man erhofft sich durch ein „easy-to-install" Packaging die LG3D Community erweitern zu können. Eine LG3D LiveCD soll es Usern ermöglichen, Looking Glass auszuprobieren, ohne es

[3] Widgets sind kleine Programme, die es dem User erlauben, schnell und kompakt Informationen abzurufen (z.B. Real-Time Wetterinformationen) oder häufig benutze Tasks schnell zu erreichen. Diese Widgets können vom User auf einem sogenannten Dashboard abgelegt werden. (http://www.apple.com/macosx/features/dashboard/)

installieren zu müssen. [6] Die Version 1.0 wurde planmäßig Ende des Jahres 2006 fertig gestellt.

Es gibt für Looking Glass keine öffentlichen Publikationen über die im Rahmen dieses Projektes durchgeführten Usabilitytests. Die Einhaltung der Design Guidelines kann nur durch laufende Usabilitytests gewährleistet werden. Die Fortschritte und rege Beteiligung der Community an der Entwicklung lassen auf positives Feedback von Testusern schließen.

Scene Manager

Der Scene Manager ist das vom Looking Glass Team entwickelte „Herzstück" von LG3D und ersetzt den klassischen Window Manager. Der Scene Manager ist das Verbindungsglied zwischen Applikationen und User. Er ist verantwortlich dafür, die Programmfenster im 3D Raum zu platzieren und zu arrangieren. Natürlich für den User schön anzuschauen und möglichst sinnvoll und ohne dabei die Usability der Anwendung zu beeinträchtigen Programmfenster sind hier nicht einfache Fenster sondern werden zu 3D Objekten. Sie können dementsprechend auch gedreht und gewendet werden oder zum Beispiel auf der Rückseite mit Notizen versehen werden bzw. könnten Konfigurationsfenster auf der Rückseite geöffnet werden (siehe Anhang, Screenshot 4). [6]

Abbildung 5: Überblicksansicht des 360° Panoramas des Scene Managers [6]

Der Scene Manager bietet eine 3D 360° Panoramaansicht und stellt dem User damit die Möglichkeit zur Verfügung, mehrere Desktops quasi übergangslos zu verwalten. Es können beliebige auf 3D gerenderte Hintergründe verwendet werden und man kann sich auch das gesamte Panorama im Überblick anzeigen lassen (Abbildung 5). [6, 28]

Damit der Platz optimal ausgenutzt werden kann, können Programmfenster zur Seite gedreht werden. Der User kann Programmfenster, die er gerade nicht braucht, einfach seitlich „abstellen". Wie Bücherrücken haben diese seitlich eine Aufschrift, eine Aufschrift, die erkennen lässt, was in dem Programmfenster geöffnet ist. Im so genannten *Bookshelf View* (siehe Anhang, Screenshot 2) ist es auch möglich, alle Fenster auf einmal zu Seite zu stellen. [6, 28]

Features

Looking Glass bietet eine Reihe von 3D Features. Viele wurden von Open Source Communitymembern entwickelt. Dazu wurde das Incubator Projekt[20] gegründet, welches der Community eine Plattform zur Verfügung stellt, um von ihnen entwickelte Applikationen zu hosten bevor diese in Looking Glass integriert werden. [6]

Incubator Applikationen

Diese in der Open Source Community entwickelten Applikationen sind als work-in-progress zu sehen, zeigen aber interessante Ideen, den Scene Manager von Looking Glass zu erweitern.[6]

Der *Hintergrundmanager* zum Beispiel ordnet Thumbnails der verfügbaren Hintergrundpanoramabilder Platz sparend in einem Kreis im 3D Raum an und erleichtert die Auswahl für den User (siehe Anhang, Screenshot 3). Der *Zoetrope Image Viewer* ordnet ebenfalls Thumbnails der Bilder im 3D Raum ähnlich einem großen Rad an und zeigt das aktuelle Bild in einem großen Fenster an. Eine der spektakulärsten Applikationen ist der *CosmoSchedulerD*. Diese unkonventionelle Terminplanungsapplikation bildet den Terminplan auf ein 3D Sonnensystem ab. Jeder Terminplan ist ein Orbit und jeder konkrete Termin wird als Planet dargestellt. Je näher der Termin rückt, desto weiter rückt er auf dem Orbit und umso größer wird der Planet (siehe Anhang, Screenshot 5). Die Entwickler wollten den Spaßfaktor einer solchen Applikation erhöhen und näherten sich dem Problem der Terminplanung in 3D von einer völlig neuen und interessant anderen Seite. [6, 20]

Zur Usability dieser Anwendungen gibt es keine veröffentlichten Publikationen oder Studien.

LINUX UND 3D

Während die Windowsgemeinde mit Vista erste Einblicke in einen dreidimensionalen Desktop erhält, gibt es unter Unix viele Implementierungen. Compiz[16], Beryl[14], Metisse[3], 3D-Desktop[11], um nur einige wenige zu nennen. Die meisten Implementierungen dieser Art bieten mehr als nur 3D Desktops und lassen dem User viel Freiraum um mit den Features auf ihrem Desktop zu „spielen".

Compiz und Beryl

Compiz und Beryl sind zwei OpenGL basierte Window- und Überlagerungsmanager[4] (composite manager) für Linux. Compiz wurde ursprünglich von Novell entwickelt und wird zum Beispiel von openSUSE [26] eingesetzt, welches 2005 von Novell etabliert wurde, um eine breitere Anwendergruppe für Linux zu interessieren. Durch eine Abspaltung von Compiz entstand Beryl, ein weiterer Window Manager, der größere Änderungen, die nur mehr

[4] Ein Überlagerungsmanager ermöglicht die Überlagerung von Programmfenstern mit anderen Grafiken, um z.B. einen Schatten darzustellen oder echte Transparenz. [27]

sehr schwer in das Original-Compiz einzubringen gewesen wären, implementieren sollte. Die Version 0.1.0 von Beryl wurde im September 2006 released. [14] Compiz und Beryl sind Komponenten- bzw. Plugin-basiert aufgebaut, d.h. jede Funktionalität und jedes Feature ist ein eigenes Plugin. Das macht es für Community-Entwickler leicht erweiterbar. 2007 wurde die Wiedervereinigung von Compiz und Beryl angekündigt. Die zwei Projekte verwendeten denselben Core Code und dieselben Plugins und werden nun unter dem vorläufigen Projektnamen *Composite Community* als ein Projekt fortgeführt. [16]

Features

Sowohl Compiz als auch Beryl bieten dem User sehr viele verschiedene Effekte, überwiegend von der Community entwickelt, die keine Usabilitytests durchgeführt zu haben scheint. Compiz schreibt zumindest auf seine Homepage, dass die Produktivität und das Vergnügen gesteigert werden[16]. Die meisten Plugins dienen hauptsächlich zur Unterhaltung der User, wie zum Beispiel das *Snow*-Plugin, bei dem es auf dem Bildschirm zu schneien beginnt oder das *Rain*-Plugin, welches Regentropfen auf dem Bildschirm simuliert. Das *Wobbly*-Plugin kann Programmfenster verzerren (siehe Anhang, Screenshot 6). Den Programmfenster-Switcher, der live-updating Thumbnails der offenen Programmfenster beim Switchen zeigt – und der neu in Windows Vista ist – gibt es in Beryl und Compiz schon lange.

Abbildung 6. Das Cube Plugin zusammen mit dem 3D Plugin von Compiz in Aktion. [16]

Das einzige 3D Feature von Compiz und Beryl ist das *Cube*/*Desktop Cube* Plugin. Dabei werden multiple Desktops auf die Seiten eines Würfels gerendert, ähnlich dem Würfel im Multiple Dashboard Patent von Apple[13]. Hier wird die extreme Modularität von Compiz und Beryl deutlich. Mit dem Cube/Desktop Cube Plugin allein kann man nichts anfangen. Um den Würfel drehen und damit zwischen den verschiedenen Desktops hin und her switchen zu können, braucht man das *Rotate*/*Rotate Cube* Plugin. Beryl bietet die Möglichkeit, die Seiten des Würfels durchsichtig zu machen, so dass man alle Seiten gleichzeitig sehen kann. Der Würfel kann aber auch „aufgefaltet“ werden und alle vier Workspaces nebeneinander angezeigt werden. Schließlich ist es noch möglich, sich die Workspaces statt außen, innen auf den Würfelseiten anzeigen zu lassen. [14, 16, 26]

Beryl und Compiz bieten jeweils mit dem *3D Plugin*/*3D Effects* Plugin noch eine kleine Draufgabe in Sachen 3D. Damit heben sich beim Rotieren des Würfels die Programmfenster von der Oberfläche ab und zeigen so Überlappungen und Ordnung der geöffneten Programmfenster (

Abbildung 6). Weiters sind Plugins verfügbar, um Programmfenster transparent zu machen oder, vergleichbar mit Apple’s Exposé, um alle offenen Programmfenster in der Übersicht zu sehen. Ein *Peel*-Plugin ermöglicht ein Programmfenster wie einen Aufkleber „abzuziehen“ und den darunter liegenden Desktop zu sehen. [14, 16]

Metisse: Forschung statt Spielereien

Metisse[3] ist eine vom Projekt *in-situ*[19] entwickeltes Window System, um neue Window Managementtechniken zu implementieren und zu testen. Man möchte sowohl Entwicklern die Möglichkeit geben, mit neuen Techniken zu experimentieren, als auch das System stabil und robust genug gestalten, um es in der täglichen Arbeit einsetzen zu können. Mit Metisse sollen keine weiteren „Spielereien“ für den Desktop entwickelt werden, sondern es zielt darauf ab, Konzepte zu entwickeln, die die Arbeit am Computer wirklich effizienter gestalten. Obwohl es auch 3D Effekte enthält, beschränkt sich Metisse nicht darauf. Viele der Effekte sind neue Fenstermanipulationsmöglichkeiten und sollen den User bei der Erfüllung täglicher Aufgaben wirkungsvoll unterstützen. Aufbauend auf Metisse wurde User Interface Façades [8] entwickelt, die die Möglichkeiten der Fenstermanipulation weiter ausnutzen.

Wie Looking Glass bietet Metisse die Möglichkeit, Programmfenster so auf die Seite zu drehen, als ob sie in den Raum stehen würden. Genauer gesagt können Programmfenster beliebig um ihre X- und Y-Achse gedreht werden.[3] Einzigartig bei diesem Window Manager ist, dass die Fenster, egal in welcher Position sie sich befinden oder wie sie gedreht sind, wie „normale“ Fenster manipuliert werden können. Ebenso, wenn Metisse im Pager Mode läuft. Der Pager Mode ist, wieder ähnlich zu Apple’s Exposé, eine Übersicht über alle offenen Fenster auf allen Desktops. Der Desktop wird dazu ausgezoomt in neun Felder eingeteilt, jedes enthält einen Desktop. Die Manipuliationsmöglichkeit scheint hier jedoch wenig Vorteil zu bringen, es sein denn, man hätte einen sehr großen Bildschirm zur Verfügung. Zum Beispiel für das neu laden von Homepages oder ähnliches könnte es sinnvoll sein. Die Programmfenster werden auch im Pager Mode in ihren aktuellen Positionen dargestellt. Auch, wenn sie transformiert sind (**Abbildung 7**). Programmfenster können maßstäblich verkleinert werden, um Platz auf dem Desktop zu sparen, oder auch dupliziert werden.

Überlappende Fenster können zurückgefaltet werden, um darunter liegende anzuzeigen. [3] Der Einsatz von Metisse bei der täglichen Arbeit hat gezeigt, dass es sinnvoll ist, dem User einen Shortcut zur Verfügung zu stellen, um ein transformiertes Programmfenster schnell wieder in die Normalposition zu bringen. So kann der User mit einem Rechts-Klick auf die Titelleiste des Fensters die Transformation rückgängig machen. [3]

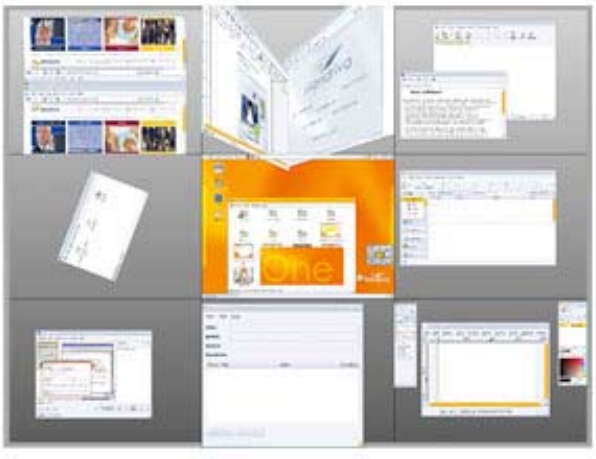


Abbildung 7. Metisse im Pager Mode. [22]

Bis jetzt haben die Entwickler Metisse noch keine Ergebnisse von Usabilitytests veröffentlicht, um die Nützlichkeit entwickelten Konzepte zu beweisen, wollen dies aber nachholen. [3] Ein neues Feature zum Copy-and-Paste zwischen sich überlappenden Fenstern wurde auf der CHI 2007[5] vom Metisse-Team präsentiert. Dabei wird das Fenster, von dem kopiert werden soll, beim Selektieren hervorgehoben und die Ecken überlappender Fenster werden automatisch quasi auf die Seite weggefaltet wie Papier. Nach Beendigung des Selektierens rollen sich die anderen Fenster wieder zurück und das hervorgehobene Fenster wird wieder an seine ursprüngliche Position zurückgestellt. Damit kann die Zeit, die für eine Copy-und-Paste Aktion benötigt wird, im Vergleich zu anderen Techniken wesentlich verkürzt werden. [4]

Metisse ist im neuen Linux der französischen Firma Mandriva, Mandriva Spring 2007, enthalten. Mandriva warnt jedoch ausdrücklich davor, Metisse in kritischen Systemen einzusetzen, da es ein Forschungssystem ist und die hundertprozentige bugfreie Funktionalität nicht gegeben ist. [22]

BUMPTOP: DER „REALE“ DESKTOP IM COMPUTER

Der Trend in der Weiterentwicklung des Interfaces geht klar in Richtung 3D und weg von der klassischen Metapher des Desktops. Dass dies nicht so sein muss, zeigt der Ansatz von Agarawala und Balakrishnan [1, 2]. Sie versuchen in ihrem Prototypen *BumpTop*, den physischen Desktop am Bildschirm abzubilden.

Dazu hat der User eine gedrehte Ansicht eines Desktops am Bildschirm, die rechts, links und vorne jeweils von Wänden begrenzt wird. Die 2½D Darstellung entstand aus dem Feedback der User, die in der ursprünglichen 2D Ansicht einfache Files nicht von Stapeln unterscheiden konnten. Um dieses Problem zu lösen wurde die Ansicht um 25° gedreht und Schatten hinzugefügt, um Tiefe zu erzeugen. [1] Auf Abbildung 8 ist deutlich zu sehen, dass BumpTop einen Desktop wesentlich realistischer abbildet als der herkömmliche 2D Desktop.

Abbildung 8. (a) zeigt einen typischen 2D Desktop mit angeordneten Icons, (b) hingegen zeigt einen Schreibtisch wie er in der realen Welt aussehen könnte, (c) zeigt BumpTop mit unterschiedlich angeordneten Objekten. [1b]

Die Interaktion mit dem User Interface soll soweit wie möglich der normalen Interaktion mit Dokumenten und Dingen auf einem Schreibtisch entsprechen. Files werden dafür mit physischen Eigenschaften (wie zum Beispiel Flexibilität, Gewicht etc.) versehen, um sie wirklichem Papier ähnlicher zu machen. Sie können auf dem Desktop geworfen werden und wenn sie kollidieren, prallen sie, wie auf einem richtigen Tisch, aneinander ab. Man kann Dokumente an den Wänden aufhängen oder zerknittern, um sie als „nicht so wichtig“ zu markieren. Um auch die Manipulation realistisch zu gestalten, wurde das Interface für die Interaktion mit einem Stift optimiert. [1]

Files werden zu 3D Quadern, die auf einer Achse sehr niedrig sind und rundherum texturiert sind, um ihren Typ kenntlich zu machen. BumpTop verwendet die Pile-Metapher für das Ordnen von Dokumenten. Diese können mit von den Autoren entwickelten Interaktionstechniken manipuliert werden. Piles (Stapel von Dokumenten) können damit aus einem Haufen von unsortierten, auf dem Desktop herumliegenden Dokumenten kreiert und geordnet werden. Mit der *Lasso'n'Cross*-Technik kann der User die Dokumente einfach näher zusammenrücken, unordentlich stapeln oder einen geordneten Stapel erzeugen.

[5] http://www.chi2007.org/

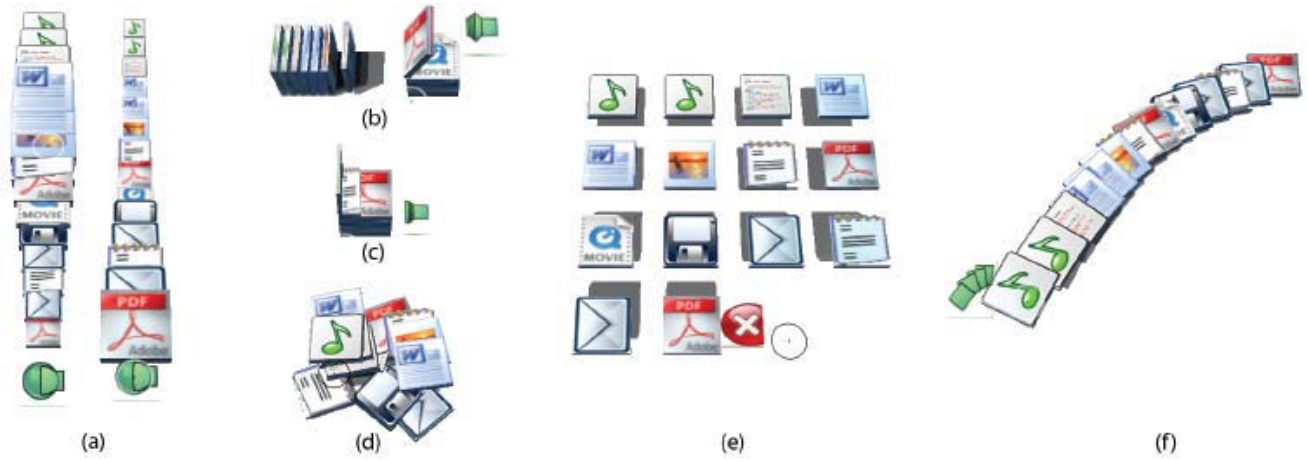


Abbildung 9. Die Manipulationswidgets von BumpTop. [1]

Dabei wird um die Dokumente, die in die Pile sollen, mit der Mouse ein Kreis gezogen und das „Create pile" Icon gekreuzt. Das Interface bietet sechs Widgets mit denen der User die Piles durchsuchen kann, die bei der Berührung einer Pile mit dem Eingabestift angezeigt werden (Abbildung 9). Das *Fisheye-Widget* (Abbildung 9 (a)) erlaubt dem User eine Pile mit einem Fisheye-View zu durchsuchen, das *Leafer-Widget* erlaubt das Durchblättern der einzelnen Dokumente eines Piles wie Seiten eines Buchs (Abbildung 9 (b)). Beim *Compression Browsing* (Abbildung 9 (c)) werden Dokumente zusammengedrückt, um die darunter liegenden sichtbar zu machen. Das *Messy/Tidy-Widget* (Abbildung 9 (d)) ordnet Dokumente wieder so an, wie sie vor dem Erstellen der Pile waren, das *Grid-Layout-Widget* (Abbildung 9 (e)) ordnet die Dokumente eines Piles am Raster nebeneinander an und das *Fan-out-Widget* (Abbildung 9 (f)) breitet die Dokumente entlang eines vom User gezeichneten Pfades aus. Um ein Dokument zu einer Pile hinzuzufügen, kann es entweder „darauf geworfen" werden, oder an einer beliebigen Stelle eingefügt werden kann, die mit dem Eingabestift ausgewählt werden kann *(Drag'n'Cross)*. Wird diese Technik zusammen mit einem der obigen Widgets verwendet, wird das Dokument an der Stelle eingefügt, an der der User sich vorher beim Suchen befunden hat. Kürzlich eingefügte Elemente stehen ein bisschen hervor, um den User weiteres Ordnen zu erleichtern. [1]

Der Prototyp wurde an sechs Usern getestet. Es waren von Neulingen bis zu Profianwendern (hinsichtlich der Eingabetechnik mit einem Stift) unter den Testpersonen. Die Ergebnisse bestätigen das Design von BumpTop. Den meisten Testpersonen gelang es, Funktionalitäten selbst zu entdecken und zu schnell beherrschen. Besonders das Werfen von Dokumenten wurde als positiv empfunden, da es dem realen Umgang mit Dokumenten entspricht. 88% der Tasks konnten ohne intensive Trainingsphase von den Usern erfüllt werden. Das Interface und die Interaktionsmöglichkeiten wurden von einigen Testusern von selbst spielerisch erlernt und viele bezeichneten es als lustig und zufrieden stellend. Mit den neu entwickelten Interaktionstechniken Lasso'n'Cross beziehungsweise Drag'n'Cross kam es durch ungewolltes aufrufen und ausführen der Widgets zu Problemen. Feedback zeigte, dass Dokument-Metadaten (z.B. die Größe des Files) besser ausgenutzt werden sollte. Man könnte zum Beispiel größere Files durch langsamere Bewegung beim Drag-and-Drop „schwerer" erscheinen lassen. [1] Große Mengen von Dokumenten bereiten ebenfalls Probleme, wie es bei der Pile Metapher bereits bekannt ist [10].

DIE MEINUNG DER USER

Trotz laufender Entwicklung und immer neuen 3D-Effekten für den klassischen Desktop wurde noch kein empirischer Beweis angetreten, dass diese Neuerungen auch wirklich sinnvoll sind. Sehr viele Projekte beschränken ihre verwendeten 3D-Effekte auf das Switchen zwischen virtuellen Desktops wie Compiz[16], Beryl[14], 3D-Desktop[11] oder Yod'm 3D[15]. Ein Vorteil der 3D-Methode gegenüber der Thumbnail-Darstellung virtueller Desktops ist die Größe. Diese 3D Animationen verwenden den ganzen Displayspace, um zum Beispiel den rotierenden Würfel mit Desktops auf jeder Seite zu zeigen. Dadurch sieht der User beinahe in Originalgröße, was auf welchem Desktop liegt und findet das gesuchte Item bzw. den richtigen Desktop schneller. Ein Nachteil könnte sein, dass der User nur für das Switchen der Desktops von einem 2D Desktop mit normaler Navigation in eine 3D-Animation wechseln muss, wo die Navigation wenig intuitiv ist. [9]

In [5] wurde eine Befragung von Testusern über 3D Effekte am Desktop durchgeführt und so wenigstens Information über die Meinungslage gegenüber dieser Entwicklung eingeholt. Es wurde zwischen den Benutzergruppen *Standardbenutzer* und *Profi* unterschieden und in jeder Gruppe 20-30 Personen befragt. Den Usern wurden 3D Desktopeffekte vorgestellt und sie danach nach ihrer Meinung gefragt. Beinahe die Hälfte (45,28%) der Testpersonen lehnten die Technologien ab, begeistert waren nur etwa ein Viertel (26,42%). Einbußen in der Performanz,

Verlust der Übersicht und Navigationsschwierigkeiten wurden als Gründe für die negative Reaktion genannt. Einige Profinutzer, die bereits 3D-Technologien am Desktop im Einsatz haben, konnten die Befürchtung von Performanzverlust bestätigen. Die Angst vor Navigationsschwierigkeiten wurde von manchen Usern darauf begründet, dass sie auch mit 3D-Spielen nicht gut umgehen könnten. Ein definitives Problem, so die Befragten, seien auch die Input Devices, die für 2D Interfaces ausgelegt seien. Ein weiterer Grund für die mangelnde Begeisterung ist die noch geringe Verbreitung der 3D-Technologien in Betriebssystemen, und dass User diese deshalb noch nicht selbst ausprobieren konnten. Als störend bei den vorgestellten 3D-Desktops wurde die Unruhe empfunden, die durch die dreidimensionalen Konstrukte auftrat (vor allem in Looking Glass), der große Platzverbrauch durch die dreidimensionale Darstellung von Elementen und unnötig überladene Elemente wie zum Beispiel zu viele, zu bunte Fenster bei mehreren geöffneten Tasks oder Transparenzeffekte (ebenfalls vor allem bei Looking Glass). 3D Darstellungen helfen zwar beim schnellen Verständnis des Elementes, sind aber für regelmäßiges Arbeiten ungeeignet. [5]

Bei der Befragung wurden durch die (potentiellen) zukünftigen Nutzer der 3D-Technologien eine Reihe von Kriterien festgelegt, die diese Technologien erfüllen müssen, sollten sie in zukünftige Betriebssysteme stärker integriert werden. Das wichtigste Kriterium war für Standardbenutzer wie für Profis die Einfachheit der Interaktion mit dem Interface. Die in 3D häufig konstruierten „Spielereien“[6] sind nicht intuitiv und verlangen dem User viel Zeit und Aufwand ab, sie zu verstehen. Auf keinen Fall sollten die 3D Effekte die Performanz des Rechners drücken, da dies die Effizienz der Arbeit beeinträchtigen kann. Außerdem muss die Verwendung der 3D Effekte einen Zusatznutzen für den User zum bisherigen 2D Interface bieten, um den Aufwand zu rechtfertigen. Das letzte identifizierte Kriterium war die Produktivität/Arbeitsgeschwindigkeit, die durch die 3D Effekte nicht beeinträchtigt werden dürfte. Die Features („Spielereien“) könnten die Aufmerksamkeit des Users zu sehr auf sich ziehen und ihn von der Arbeit abhalten. [5]

Als letzter Punkt der Befragung sollten User angeben, warum sie den 2D Desktop schätzen. Hier rangierten Schnelligkeit, Sichtbarkeit aller Icons auf einen Blick und die Personalisierung des Desktops durch Hintergrundbilder, Design oder Anordnen der Icons auf den ersten Plätzen.[5]

FAZIT

An den obigen Beispielen wird deutlich, dass 3D Desktop Effekte in die modernen Betriebssysteme Einzug gehalten haben, wenn auch nicht übermäßig ausgeprägt. Zu oft wurde leider kein wissenschaftlicher Beweis veröffentlicht, der den Sinn oder den Nutzen der Effekte beweist. Nicht von der Hand zu weisen ist aber, dass die Community sich rege bei der Entwicklung oben genannter Projekte [11, 14, 16, 20] beteiligt und diese weiterentwickelt. Während Windows und Mac OS X User sich noch gedulden müssen, haben Linux-User schon länger die Möglichkeit, sich selbst ein Bild der Features von 3D Window Managern zu machen. Obwohl die Entwicklung immer mehr in Richtung 3D Desktops geht, ist die allgemeine Meinung doch eher negativ [5] und es bleibt abzuwarten, wie weit 3D Desktop-Effekte sich in kommerziellen Betriebssystemen durchsetzen werden. Bisher sehen die User nicht sehr viel Nutzen in dieser Entwicklung und vertrauen auf die Vorteile des bekannten 2D Desktops.

Vor allem bei Microsoft und Apple hat man das Gefühl, dass sie sich nicht so recht „trauen“, dem User nicht zuviel zumuten wollen. Da Windows schon seit der Task Gallery Research im Bereich 3D-Interfaces betreibt und in [23] als Ergebnis eine Präferenz der User in Richtung 3D festgestellt hat, hätte man bei Vista mehr 3D Features erwartet. Es bleibt ein glatter Widerspruch: Man wollte mit Task Gallery zeigen, dass ein 3D Umfeld den User besser bei seiner Arbeit unterstützt als der herkömmliche 2D Desktop. In der Umsetzung wird diese „Erkenntnis“ jedoch vollkommen vernachlässigt. Sehr viel sinnvoller ist hier schon der Exposé-Effekt von Apple. Der User kann schnell zwischen Programmfenstern wechseln und auch alle offenen Applikationen auf einen Blick sehen. Apples 3D-Cube Effekt zum schnelleren Switchen von Useraccounts ist vermutlich für User weniger hilfreich, als einfach nett anzusehen. Man kann annehmen, dass auf einem privaten Computer der User nicht so oft schnell gewechselt werden muss. In einem Büro, in dem der Rechner von mehreren Usern genutzt werden soll, macht es durchaus Sinn. Ob man dazu allerdings einen 3D-Effekt braucht, ist fraglich. Ob die Implementierung von Apples patentiertem 3D-Dashboard einen Einfluss auf die Produktivität hat, bleibt abzuwarten. Der Ansatz erscheint sinnvoll, da auch viele Steuerungswidgets Platz am Desktop wegnehmen können und so der Zeitgewinn beim Verwenden dieser durch den verbrauchten Desktopplatz wieder relativiert wird.

Suns Looking Glass[28] strebt ein komplettes Neudesign des Desktops mit 3D-Anreicherungen an. Die Taskleiste sowie Icons sind als 3D Grafiken dargestellt. Geöffnete Programmfenster werden zu dreidimensionalen Objekten, die sowohl vorne als auch auf der Rückseite genutzt werden können. Interessant ist die Darstellung eines größeren Desktopspaces als 360° Panoramaansicht eines Ortes, womit mehrere virtuelle Desktops quasi zu einem großen verschmelzen. Dies scheint eine innovative Idee zu sein, lässt möglicherweise den User jedoch schnell den Überblick verlieren. Das Interface ist sehr farbenfroh und setzt vor allem auf die Wirkung von Hintergrundbildern für den Scene Manager. Aber auch Sun bleibt Beweise für die

[6] Spielereien sind in diesem Zusammenhang grafisch beeindruckende, aber nicht sinnvolle oder undurchschaubare 3D-Features.[5, S 118]

Verbesserung der Effizienz oder Produktivität durch das Verwenden dieses Window Managers schuldig. Das es durch seine beeindruckende Grafik und Effekte sicherlich einen höheren Spaßfaktor bei der Verwendung erzielt als herkömmliche 2D-Desktops, ist sicher.

Die beispielhaft vorgestellten Linux Window Manager Compiz und Beryl glänzen durch eine Fülle von „Spielereien", die dem User wahrscheinlich die Zeit vertreiben, aber ebenso wahrscheinlich auch nichts mit verbesserter Usability oder gar Produktivitätssteigerung zu tun haben. Für den rotierenden Würfel als Desktop-Switcher gibt es gewisse vorteilhafte Argumente [9], jedoch rechtfertigt dieses eine Feature nicht den durch den Einsatz des Window Managers zu erwartenden Performance-Verlust des Rechners. Metisse hingegen trumpft zwar auch mit vielen Effekten auf, widmet sich doch der Erforschung und Verbesserung des Desktops und des Userverhaltens in der Interaktion mit diesem und wird hoffentlich in Zukunft Ergebnisse und Beweise für den Nutzen auch von 3D Effekten am Desktop bringen.

Ein besonders interessanter weil gänzlich neuer Ansatz im Bereich 3D Desktop ist in [1, 2] beschrieben. Dabei wird die Desktop-Metapher – anstatt durch 3D Effekte ersetzt – wörtlich genommen und eine 2½D Sicht auf einen virtuellen Schreibtisch mit allen physikalischen Merkmalen als Desktop verwendet. Testpersonen dieses Prototyps haben durchgehend positiv auf diesen neuen Ansatz reagiert. [1] Das lässt die Vermutung zu, dass neue Desktops sich in diese Richtung entwickeln können. Ein Problem sind große Mengen an Files, für die diese Metapher nicht funktioniert, zu deren Produktion User aber neigen.

Ein umfassender Überblick über laufende Projekte im Bereich 3D ist unter [29] zu finden. Viele der Links sind nicht mehr up-to-date, die meisten genannten Projekte aber im Internet zu finden.

REFERENZEN

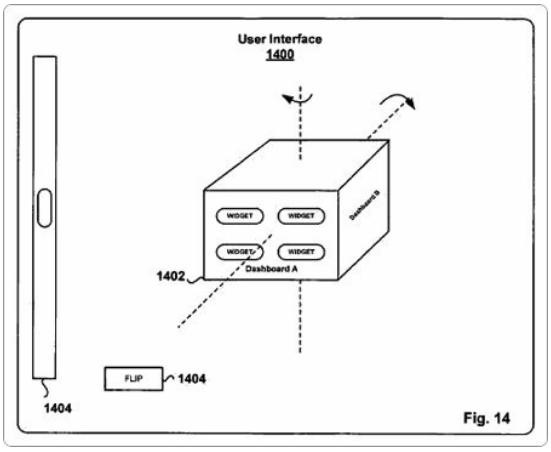
1. Agarawala, A., Balakrishnan, R., Keepin' it Real: pushing the Desktop Metaphor with Physics, Piles and the Pen. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1283–1292.
2. Agarawala, A., Enriching the Desktop Metaphor with Physics, Piles and the Pen, *Master Thesis*, University of Toronto, 2006.
3. Chapuis, O., Roussel N., Metisse is not a 3D desktop!. In *Proc. of the 18th annual ACM symposium on User interface software and technology 2005*, ACM Press (2005),13-22.
4. Chapuis, O. and Roussel, N., Copy-and-Paste Between Overlapping Windows. In *Proceedings of ACM CHI 2007 Conference on Human Factors and Computing Systems*, pages 201–210, April 2007. ACM Press (2007).
5. Graf, M., Die Desktop-Metapher – eine empirische Studie und Implikationen, *Diplomarbeit*, Universität Klagenfurt, Österreich, März 2007.
6. Kawahara, H., Byrne, P., Johnson, D., Gadepalli, K., Project Looking Glass: A Comprehensive Overview of the Technology, *Sun Microsystems*, März 2006.
7. Robertson, G., van Dantzich, M., Robbins, D., Czerwinski, M., Hickley, K., Risden, K., Thiel, D., Gorokhovsky, V., The Task Gallery: a 3D window manager. In *Proc. of the SIGCHI conference on Human factors in computing systems 2000*, ACM press (2000), 494 - 501.
8. Stuerzlinger, W., Chapuis, O., Philips, D., Roussel, N., User interface façades: towards fully adaptable user interfaces. In Proc. of the 19th annual ACM symposium on User interface software and technology 2006, ACM Press (2006), 309-318.
9. Tomitsch, M., Trends and Evolution of Window Interfaces. *Diploma Thesis*, University of Technology, Vienna, December 2003.
10. Whittaker, S., Hirschberg, J., The Character Value and Archive of Personal Paper Archives, ACM Transactions on Computer-Human Interaction, 8(2), pages 150–170, ACM Press (2001).

Internetadressen

11. 3D-Desktop, http://sourceforge.net/projects/desk3d/, 7.5.2007.
12. Apple, Aqua User Interface, http://www.apple.com/macosx/overview/aquauserinterface.html, 7.5.2007.
13. Apple patent details multiple Dashboards in 3D, http://www.macnn.com/articles/07/05/03/multiple.dashboard.patent/, 7.5.2007.
14. Beryl Project, http://www.beryl-project.org, 5.5.2007.
15. Chris' Soft, Yod'm 3D, http://chsalmon.club.fr/index.php?en/Yod-m-3d-about, 6.5.2007.
16. Compiz Project, http://compiz.org/Home/Start, 5.5.2007.
17. Gottlieb-Graham, N., Vista: An open-minded, Windows-using Mac fan's perspective, http://unifiedforever.wordpress.com/2007/04/07/vista-an-open-minded-windows-using-mac-fans-perspective/, 28.4.2007.

18. Hutsko, J., A Mac user switches to Vista, http://www.msnbc.msn.com/id/16873608, 28.4.2007.
19. in-situ Project, http://ralyx.inria.fr/2006/Raweb/in-situ/uid47.html, 7.5.2007.
20. Iverson, W., Behind The Scenes of Project Looking Glass, http://today.java.net/pub/a/today/2004/07/06/3ddesktop.html, 3.5.2007
21. LG3D Incubator Project, https://lg3d-incubator.dev.java.net/, 4.5.2007.
22. Mandriva Linux, http://www.mandriva.com/projects/metisse/, 7.5.2007.
23. Microsoft Research, The Task Gallery, http://research.microsoft.com/adapt/TaskGallery/, 30.4.2007.
24. Microsoft Research, VIBE Visualization and Interaction for Business and Entertainment, http://research.microsoft.com/vibe/, 30.4.2007.
25. Microsoft, Windows Vista, http://www.microsoft.com/windows/products/windowsvista/default.mspx, 28.4.2007.
26. Microsoft, Windows Vista Features: Flip 3D, http://www.microsoft.com/windows/products/windowsvista/features/details/flip3D.mspx, 28.4.2007.
27. Novell, openSUSE Project, Compiz, http://de.opensuse.org/Compiz, 6.5.2007.
28. Sun Microsystems, Project Looking Glass, http://www.sun.com/software/looking_glass/details.xml, 28.4.2007.
29. Sun Microsystems, Project Looking Glass Related Technologies, https://lg3d.dev.java.net/lg3d-related-technologies.html, 7.5.2007.
30. Sun Microsystems, Project Looking Glass 3D Desktop Visual Design Discussions, https://lg3d.dev.java.net/lg3d-ui-design.html, 2.5.2007

ANHANG



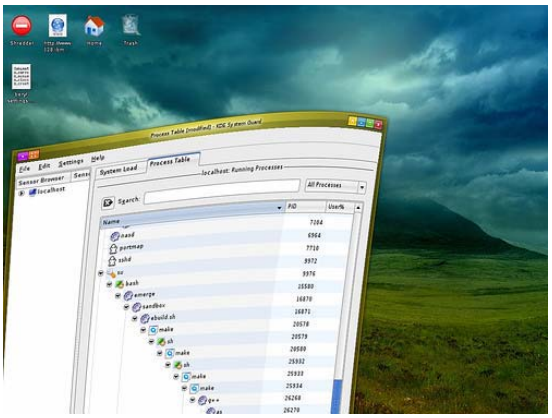
Screenshot 1. Multiple Dashboards von Apple. [13e]

Screenshot 2. Bookshelf View der Programmfenster im Scene Manager. [6]

Screenshot 3. Ein 3D Hintergrund-Manager für den LG3D Scene Manager. [20q]

Screenshot 4. Auf der Rückseite eines Programmfensters kann zum Beispiel die Konfiguration geöffnet werden. [6]

Screenshot 5: Der CosmoSchedulerD bildet einen Terminplaner auf das Sonnensystem in 3D ab. [6]



Screenshot 6. Das Wobbly-Plugin im Compiz Window Manager [16j]

Brain-Computer Interface
Wo Mensch und Maschine aufeinander treffen

Vera Koren
Studentin der Universität Klagenfurt
Adolf-Schärf-Straße 7
9030 Viktring
v1koren@edu.uni-klu.ac.at

ABSTRACT
Mensch – Computer. Auf diesen zwei „Individuen" liegt seit vielen Jahren der Fokus der Wissenschaft. Wie können Mensch und Computer zusammenspielen? Gibt es Möglichkeiten beides zu vereinen? Diese und viele andere Fragen beschäftigen Wissenschaftler seit vielen Jahren. In den frühen 1970er gelang es eine Basis zur Lösung all dieser Fragen zu finden, das BRAIN-COMPUTER INTERFACE (BCI). Seit den 1970er sind einige Jahre vergangen in denen die Wissenschaft nicht geruht hat. Der nachstehende Artikel soll nun einen Einblick in „The State of the Art" von Brain-Computer Interfaces geben.

Schlüsselwörter
Brain-Computer Interface (BCI), invasiv, nicht-invasiv

EINLEITUNG

Die Interaktion zwischen Mensch und Computer beschäftigt die Wissenschaft schon seit vielen Jahren. Es entstehen immer mehr neue Interaktionparadigmen und damit verbundene Technologien sind einer ständigen Weiterentwicklung unterworfen.

Schon seit einiger Zeit haben Forscher an einer Verbindung zwischen Mensch und Maschine gearbeitet, was sich nach einer Idee aus einem Sciencefictionfilm anhört, ist nun aber Realität geworden. Das BRAIN-COMPUTER INTERFACE, kurz BCI genannt, ist ein Interface, das Gehirnströme und andere Gehirnaktivitäten des Menschen misst und diese in digitale Signale umwandelt, wodurch sie für Computer verständlich gemacht werden. [10]

Diese neuartige Technologie wird in sehr vielen verschiedenen Bereichen zur Anwendung kommen. So zum Beispiel demonstrierte eine Gruppe von Personen in Korea ein Spiel, in dem der User eine Figur durch ein Labyrinth navigieren musste, in dem er sich die Hand- und Beinbewegung der Figur vorstellte. [8]

Wenngleich BCIs hauptsächlich für medizinische Rehabilitationen Verwendung finden werden, stellt es trotzdem einige aufregende Gelegenheiten für zukünftige Videospiele dar. [8]

Mit Hilfe von BCIs wird es Menschen, die ihre Arme und Hände nicht normal bewegen können, möglich sein, alleine durch Gehirnströme bzw. Gehirnaktivitäten, Computer, Rollstühle, und andere Dinge zu benutzen und zu steuern. [10]

Obwohl der Gedanke, dass ein solches System in der Realität zur Anwendung kommt für die meisten Menschen noch sehr fremd ist, gibt es bereits den ersten funktionierenden Prototyp, der die Technologie eines BCI – Brain-Computer Interfaces implementiert. Bei diesem Prototyp handelt es sich um einen gehirngesteuerten Rollstuhl. [12] Wenn dieser Rollstuhl, den Tests standhält, dann ist dies der erste und bedeutendste Schritt für Menschen mit einer körperlichen Behinderung, in Richtung ein- wenigstens zum Teil - selbständiges Leben.

Die Zielsetzung der Arbeit liegt in der Erörterung des State of the Art von Brain-Computer Interfaces.

Der Artikel basiert auf drei grundlegenden Fragen:

- Was versteht man unter dem Begriff Brain-Computer Interface und wie funktioniert es?
- Welche Benutzungsprobleme werden mit Hilfe des BCI gelöst?
- Welche Einsatzmöglichkeiten bietet ein BCI Menschen mit / ohne Behinderungen?

BCI – WAS IST DAS?

Ein BCI erlaubt die direkte Kommunikation zwischen Mensch und Maschine, ohne die Verwendung von muskularen Aktivitäten. [5]

Es handelt sich dabei um ein Gerät, das es Menschen erlaubt, durch die Verwendung von Gedanken bzw. Gehirnströmen einen Computer oder andere elektronische Vorrichtungen zu steuern. Ein BCI benützt die

elektrophysiologischen menschlichen Signale des Elektroenzephalogramms (EEG), die vom menschlichen Gehirn produziert werden, um zu kommunizieren oder verschiedene Geräte zu steuern. Diese Signale werden durch den Computer verstärkt, digitalisiert und manipuliert und zu Aktionen oder Befehlen „übersetzte“. Der Manipuliationsprozess und der Übersetzungsprozess sind keine einfachen Aufgaben. Es handelt sich dabei um einen Prozess, der aus vielen Teilaufgaben zusammengesetzt ist, welche sich in drei Hauptprozesse einteilen lassen[9]:

- Filterung des Signals (unerwünschten Daten werden aus den Daten gefiltert)
- Kategorisierung oder Feature Extraction Prozess
- Umwandlung der Eigenschaften vom Signalisieren durch den Übersetzungsprozess in Symbole bzw. Befehle

Abbildung 1 verdeutlicht die Architektur eines BCI noch einmal.

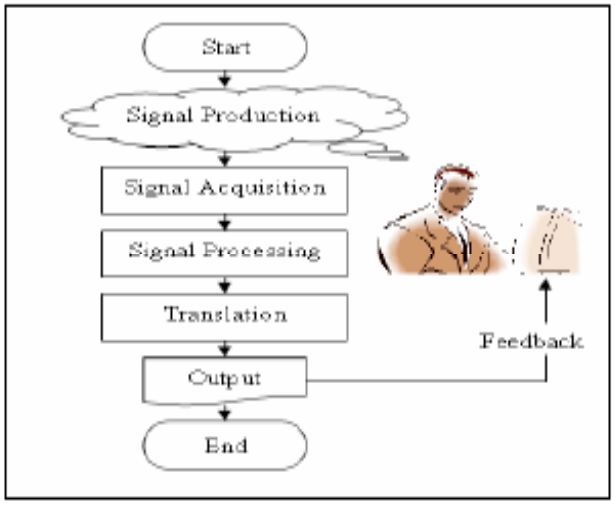


Abbildung 1: Architektur eines BCI [9]

Im Allgemeinen bieten BCIs für Menschen mit Behinderungen, neuromuskularen Störungen sowie Rückenmarksverletzungen eine Erleichterung der Kommunikation. Abgesehen vom medizinischen Anwendungsbereich, finden BCIs mittlerweile auch bei aufregenden Multimediaanwendungen Verwendung. Bei diesen Multimediaanwendungen handelt es sich zum Beispiel um ein Geschicklichkeitsspiel, bei dem das neue Niveau der Steuermöglichkeiten auch für gesunde Menschen verwendet wird. [6]

BCI – GESCHICHTE

In den Frühen 1970ern begannen Wissenschaftler erstmals aktiv mit dem BCI zu forschen. 1973 baute Jacques Vidal, heutiger Professor an der Universität in LA - Kalifornien, das erste BCI. Er leitete zu dieser Zeit auch das staatliche geförderte BCI Projekt an der Universität in LA. In diesem Projekt haben Wissenschaftler mit dem Implantieren von einfachen BCI Sensoren bei Ratten, Mäusen, Affen und auch schon bei Menschen experimentiert. [9,10]
Wegen der langen Antwortzeiten, der überdimensionalen Fehlerhäufigkeiten, hohen Kosten und langen Lernzeiten, fand die Technologie fand im realen Leben keine Anwendung. [9]

In den späten 1990ern, zeigten Forscher des Georgia Institute of Technology und der Empory Universität, welches enormes medizinisches Potenzial in den BCIs steckt. Es wurde einem Patienten, der durch einen Unfall vom Hals abwärts gelähmt war und nicht sprechen konnte, Elektroden in die motorische Hirnrinde implantierten. Durch die Technik wurde es dem Patienten ermöglicht mittels Bewegungen eines Computercursors mit seiner Umwelt zu kommunizieren. [10]

Im Jahr 1999 gelang es Wissenschaftlern an der MCP Hahnemann School of Medicine und dem medizinischen Center der Duke Universität, Ratten so zu trainieren, dass diese durch Verwendung ihrer Hirnsignale in der Lage waren, einen roboterähnlichen Wasserspendenden Arm zu steuern. [10]

BCI - ARTEN

Jede Aktivität des Gehirns, egal ob es sich dabei um Sinnesreize, die Steuerung einer Bewegung oder nur die Vorstellung an die Bewegung eines Körperteils handelt, führt zur Aktivierung des motorischen Kortex. Dies hat zur Folge, dass auf der Kopfhaut schwache Spannungsschwankungen entstehen, die gemessen werden können. [14]

Prinzipiell gibt es zwei verschiedene BCI Ansätze zur Abnahme von Gehirnströmen. Die Ursprünge dieser beiden Ansätze liegen an der Universität von Süd Kalifornien, deren Professor Theodore Berger den Vorsitz der World Technology Evaluation Center’s Panel on Brain Computer Interfaces Inne hat.[10]

Auf der einen Seite wäre da der invasive Ansatz, bei dem Elektroden direkt ins Gehirn des Patienten implantiert werden. Auf der anderen Seite der nicht-invasive Ansatz, bei dem medizinisches Scanning und Sensoren auf Kappen oder Kopfbändern angebracht sind, die die Gehirnströme messen. [10]

Invasive Ansatz - Allgemein
Der invasive Ansatz bringt einige Probleme und Nachteile mit sich. Diese Probleme sind verschiedenster Natur, angefangen bei möglichen medizinischen Problemen, über ethnische Probleme bis hin zu Performance Problemen. [7]

Wie schon zuvor erwähnt, werden beim invasiven BCI Elektroden direkt am Gehirn des Patienten angebracht. Dies kann ausschließlich durch eine Operation durchgeführt werden und birgt somit ebenso Gefahren, wie auch jede anderwärtige Implantation. Es kann also zu Infektionen, Abstoßungserscheinungen und auch Hirnschäden kommen. [10]

Es gibt aber auch einen wesentlichen Vorteil, den die Invasive Methode mit sich bringt, und zwar handelt es sich hierbei um die Signalqualität. Dadurch, dass bei der invasiven Methode die Elektroden direkt an dem Platz im Gehirn angebracht sind, an dem die Gehirnströme entstehen, ist die Qualität der Signale enorm hoch. Es wird typischer Weise immer auf einen der folgenden Methoden aufgebaut: [4]

- Elektrokortikogramm
- Aufnahme des einzelnen Neurons
- Aufnahme von Neuronenstämmen

Die Forschung auf diesem Gebiet wird aktiv ausgeübt, mit dem Ziel komplexe und exakte Bewegungen durch die Decodierung von Bewegungsinformationen aus dem dafür zuständigen Hirnbereich, wiederherzustellen. [4]

Invasiver Ansatz - Methoden
Im Folgenden wird nun ein Einblick in die wichtigste und häufigst verwendete der zuvor aufgelistete invasive Methoden, nämlich dem Elektrokortikogramm, gegeben.

Elektrokortikogramm (ECoG)
Bei einem Elektrokortikogramm, kurz EcoG genannt, handelt es sich um eine Methode der medizinischen Diagnostik zur Messung von elektrischen Signalen mit Hilfe von auf der Kortexoberfläche, also direkt auf der Hirnoberfläche, platzierten Elektroden. [17]

Die Amplituden der Gehirnströme sind ungefähr zehnmal so groß wie beim EEG, bei dem es sich um die nicht invasive Methode eines ECoG handelt, es können auch höhere Frequenzanteile nachgewiesen werden. Die größeren Amplituden beim ECoG lassen sich durch zwei wesentliche Dinge erklären. Auf der einen Seite tritt der Verlust durch die Hirnhäute, die Schädeldecke und die Kopfhaut nicht mehr auf, und auf der anderen Seite wird ein kleineres Areal als beim EEG abgeleitet, weshalb auch weniger Neuronen synchron erregt werden müssen, um ein Signal zu erzeugen.[3]

Die aus dem ECoG resultierenden Ergebnisse sind weit weniger anfällig, als Messergebnisse, die aus nicht-invasiven Messmethoden gewonnen werden. Dennoch ist Aufgrund der für das ECoG notwendigen Operation die Anwendbarkeit stark eingeschränkt. [2]

Invasiver Ansatz - Anwendung
Trotz der Probleme die der invasive Ansatz mit sich bringt, gibt es einige Forschungsprojekte, die sich mit dem Einsatz der invasiven Methode beschäftigen. Im Folgenden wird auf ein spezielles Projekt eingegangen, um einen kurzen Einblick in die invasive Methode zu bekommen.

BrainGate Neural Interface System
Die Firma Cyberkinetics Neurotechnology Systems entwickelte in Kooperation mit der Brown Universität das BrainGate Neural Interface System, kurz BrainGate genannt. Beim BrainGate handelt es sich um ein medizinisches Gerät, dass Patienten mit Rückenmarksverletzungen oder anderen Arten von Bewegungseinschränkungen die Steuerung eines Computers mittels Gedanken ermöglicht. [10]

Durch die Steuerung des Computers mittels Gedanken soll es den Patienten ermöglicht werden, Zugang zu alltäglichen Tätigkeiten zu bekommen. Diese Tätigkeiten reichen über typischen Computerfunktionen (z.B., Kommunikation) bis hin zur Steuerung der Gegenstände im Umfeld (Telefon, Fernseher, Lichter). [1]

Abbildung 2: BrainGate System [Bra1]

Um dies in die Realität um zu setzen bedarf es einer Vielzahl von Forschungsprojekten, die sich alle mit einem zentralen Thema beschäftigen: der Transfer der Gedanken in Signale die Aktionen, wie etwa das Bewegen einer Hand, hervorrufen. [10]

Das BrainGate™ System basiert auf der Plattformtechnologie Cyberkinetics, mit Hilfe derer die Neuronenströme abgefragt, übertragen, analysiert und angewendet werden. Das System besteht aus einem Sensor (Elektrokortikogramm), der auf der Bewegungsrinde des

Gehirns und dem Kortex eingepflanzt wird und die Gehirnsignale abnimmt und analysiert. [1,16]

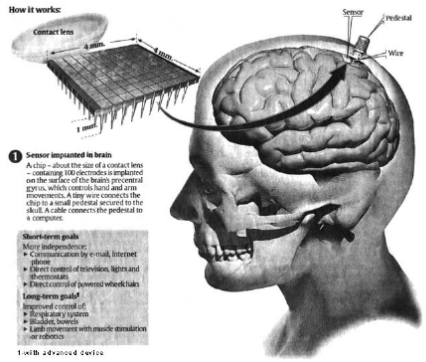

How it works:
Contact lens
4 mm
4 mm
1 mm
Sensor
Pedestal
Wire
1 Sensor implanted in brain
A chip – about the size of a contact lens – containing 100 electrodes is implanted on the surface of the brain's precentral gyrus, which controls hand and arm movements. A tiny wire connects the chip to a small pedestal secured to the skull. A cable connects the pedestal to a computer.
Short-term goals
More independence:
• Communication by e-mail, Internet, phone
• Direct control of television, lights and thermostats
• Direct control of powered wheelchairs
Long-term goals*
Improved control of:
• Respiratory system
• Bladder, bowels
• Limb movement with muscle stimulation or robotics
*with advanced device

Abbildung 3: BrainGate Sensor – Elektrokortikogramm [16]

Das Prinzip auf dem das BrainGate System basiert, baut auf der interaktiven Gehirnfunktion auf, die Gehirnsignale erzeugt, obwohl sie im Falle eines Gelähmten nicht zu den Armen, Händen und Beinen geschickt werden. Diese Gehirnsignale werden analysiert und in Cursorbewegungen übersetzt. Dadurch wird es Patienten möglich gemacht, als Alternative zur Klassischen Computer-Interaktionsform (Maus), die Gedanken zu verwenden. Das BrainGate System ist nur eine Untersuchungsvorrichtung, die bei klinischen Studien Verwendung findet und nicht für Verkauf angedacht. [1]

Sensor
Signals
Neurons
2 Sensor detects neuron activity
The electrodes sense neural impulses – the electrical signals exchanged by neurons within the brain. Those signals are sent when the brain executes a body movement.
3 Neural activity converted to output signal
The computer translates the signals into communication output, allowing a patient to move a cursor on a computer screen merely by thinking about it.

Abbildung 4: BrainGate System – Cursor [16]

Nicht-invasiver Ansatz - Allgemein

Menschliche BCI Forschung hat sich auf die nicht-invasive Methoden zur Darstellung von Gehirnaktivitäten fokussiert. [13] Diese Methode biete einige Vorteile gegenüber der invasiven Methode, da hierbei keine Implementation von Elektroden notwendig ist, werden die Nachteile der invasiven Methode gänzlich vermieden. Die nicht-invasive Methode bietet die Möglichkeit Gehirnströme an vielen verschiedenen Punkten des Gehirns zu messen, was eine breitere Analyse der Gehrinaktivitäten ermöglicht. Dies ist ein sehr großer wenn nicht sogar der bedeutendste Vorteil gegenüber der invasiven Methode, denn die Bewegungssignale, die verschiedenste Körperteile in Bewegung setzen, sind an verschiedenen Punkten im Gehirn platziert. [10]

Es gibt verschiedenste nicht-invasive Methoden um die Gehirnaktivitäten darzustellen[6]:

- Positronen-Emissions-Tomographie (PET)
- Funktionelle Magnetresonanztomographie (fMRI)
- Magnetoenzephalographie (MEG)
- Elektroenzephalographie (EEG)

Nicht-invasiver Ansatz - Methoden

Im Folgenden wird nun ein Einblick in die diversen nicht-invasiven Methoden gegeben, um den Unterschied und die Funktionsweisen dieser Methoden besser verstehen zu können.

Positronen-Emissions-Tomographie - PET

Die Positronen-Emissions-Tomographie, kurz PET genannt, ist ein Verfahren der Nuklearmedizin, das dreidimensionale Modelle und Schnittbilder von lebenden Organismen erzeugt. Mit Hilfe der PET ist es zum Ersten Mal möglich, die Funktionalitäten von menschlichen Hirnarealen zu messen, wobei der Patient bei Bewusstsein ist. [11,18]

Als Basis von PET fungiert die Darstellung der Verteilung einer radioaktiv markierten Substanz, auch Radiopharmakon genannt, im Organismus. Radiopharmaka sind Substanzen, die mit einem Radionuklid - also einem instabilen Atom, das radioaktiv zerfällt - markiert sind. In der PET werden unter anderem die radioaktive Isotope der Elemente Fluor, Kohlenstoff, Stickstoff und Sauerstoff verwendet. Mit diesen radioaktiven Isotopen lassen sich Moleküle herstellen, die der Organismus nicht von ihren nichtradioaktiven Gegenstück unterscheiden kann und die deshalb in den gewöhnlichen Stoffwechsel eingehen. Die Radionuklide, welche beim Zerfall Positronen (Antiteilchen des Elektrons) aussenden, eigenen sich besonders gut zur PET. Dabei trifft nach kurzer Distanz ein Positron auf ein Elektron, was die Vernichtung beider Teilchen und die Entstehung zweier hochenergetischer Photonen (Gammastrahlung) zur Folge hat. Diese Vernichtungsstrahlung ermöglicht den Nachweis und die Schätzung der Lokalisierung der Positronenemission. [18]

Der Patient bei dem eine PET durchgeführt werden soll, muss das Radiopharmakon per Injektion oder Inhalation verabreicht bekommen, damit dieses in den Organismus aufgenommen wird. Er wird auf einem beweglichen Tisch

so positioniert, dass der Kopf, bzw. der zu untersuchende Körperabschnitt im Sichtbereich des PET-Scanners liegt. Dieser Scanner besteht aus mehreren hundert ringförmig angeordneten Gamma-Detektoren, die in Koinzidenz geschaltet sind. Die bei der Positron-Elektron-Vernichtung entstehende Vielzahl an Gammastrahlung wird somit zur Berechnung des Schnittbildes (siehe Abbildung 5) und eines dreidimensionalen Modells herangezogen. [18]

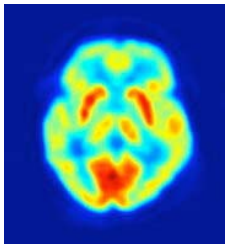


Abbildung 5: PET [18]

Es ist vom verwendeten Radiopharmakon abhängig, welche Funktionalitäten des Hirns mit Hilfe der PET grafisch dargestellt werden, denn jedes Radiopharmakon lässt andere Funktionalitäten bzw. Gehirnbereiche sichtbar werden. Zur Funktionsuntersuchung des Gehirns wird dem Patienten radioaktiver Sauerstoff (^{15}O) zur Inhalation gegeben. Dadurch lässt sich die Durchblutung des Gehirns in der PET abbilden. Höhere Durchblutung in einem Hirnareal lässt auf höhere neuronale Aktivität schließen. Ein weiteres Radiopharmakon, welches zur Untersuchung des Gehirns eingesetzt wird, ist das Fluor-Deoxyglucose (^{18}F-FDG), es dient dazu die Stoffwechselaktivität des Gehirns beurteilen zu können. [11,18]

Funktionelle Magnetresonanztomographie - fMRI

Die funktionelle Magnetresonanztomographie, kurz genannt fMRI (Functional Magnetical Resonance Imaging), ist ein Verfahren mit hoher räumlicher Auflösung zur Darstellung von aktivierten Strukturen innerhalb des Gehirns. Beim fMRI handelt es sich um eine sehr junge Weiterentwicklung der klassischen Magnetresonanztomographie, mit der es möglich ist Stoffwechselvorgänge, die durch Gehirnsaktivitäten entstehen, sichtbar zu machen. Mit Hilfe von Wahrscheinlichkeitsberechnungen, wird der Ort der Gehirnaktivität lokalisiert.[19]

Im Normalfall läuft eine fMRI Untersuchung in drei Phasen ab[19]:

1. **Prescan**, dabei handelt es sich um einen kurzen, gering auflösenden Scan, mit einer Dauer von 30 Sekunden
2. **3D MRT-Scan**, dabei handelt es sich um einen räumlich hoch auflösenden Scan, mit dessen Hilfe die Anatomie des zu untersuchenden Bereichs detailgetreu dargestellt werden kann. Die Dauer des 3D MRT-Scans beträgt zwischen 10 und 15 Minuten
3. **fMRI-Scan**, hierbei handelt es sich um einen schnellen, räumlich gering auflösenden Scan, mit Hilfe dessen die stoffwechselbedingten Unterschiede im untersuchten Gewebe registriert werden. Der fMRI-Scan dauert zwischen 6 und 7 Minuten. Bei dieser Art von Scan, muss der zu Untersuchende einem wiederholten Reiz ausgesetzt werden. Dieser Reiz wird zum Beispiel durch „Finger-Tapping“ erzeugt. Dabei muss der Patient die Finger einer Hand nacheinander zum Daumen derselben Hand bewegen. Diese Bewegung erfolgt ungefähr 30 Sekunden lang, gefolgt von einer 30 sekündigen Pause. Dieses „Finger-Tapping“ sollte ein paar mal wiederholt werden, um ein möglichst genaues Resultat zu erlangen. Die aus den Reizphasen gewonnen Daten, werden mit den Daten der Ruhephasen verglichen und der daraus resultierende Unterschied auf den vorher durchgeführten MRT-Scan als farbliche Markierungen projiziert (siehe Abbildung 2). [19]

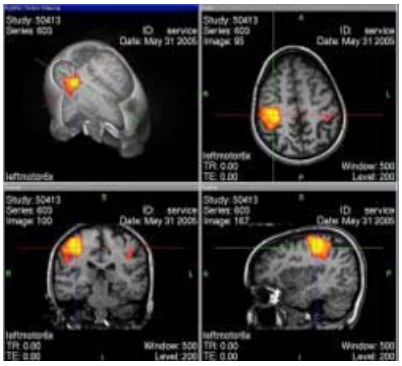


Abbildung 6: fMRI Aufnahme nach linksseitigem „Finger-Tapping“ [19]

Der farbig dargestellte Bereich im Gehirn, stellt einen erhöhten Stoffwechsel und damit eine Gehirnaktivität dar. [19]

Magnetoenzephalographie - MEG

Bei der Magnetoenzephalographie handelt es sich um ein Verfahren zur Messung der magnetischen Aktivität des Gehirns. Die Magnetfelder werden dabei durch supraleitende Spulen oder Spulensysteme erfasst und dann von äußeren Sensoren, den so genannten SQUIDs, gemessen. Bei SQUID handelt es sich um eine Supraleitende Quanteninterferenzeinheit (Superconducting QUantum Interference Device), welche ein Sensor zur sehr präzisen Messung extrem kleiner Magnetfeldänderungen ist. [20,24]

Äußere Störungen müssen bei der Anwendung von MEG möglichst vollständig abgeschirmt werden, da die magnetischen Signale des Gehirns nur wenige Femtotesla – Messgröße in der magnetische Strahlen, magnetische Flussdichte und Induktion gemessen wird – betragen. [20,21] Die magnetischen Signale entstehen durch die

elektrischen Ströme von aktiven Nervenzellen im Gehirn. Mit Hilfe des MEG können somit Daten aufgezeichnet werden, die ohne zeitliche Verzögerung einen aktuellen Überblick über die Gesamtaktivität des Gehirns gibt. [20]

Elektroenzephalographie - EEG

Bei der Elektroenzephalographie, kurz EEG genannt, handelt es sich um eine Methode der medizinischen Diagnostik zur Messung der elektrischen Gehirnaktivitäten, die anhand der Spannungsschwankungen an der Kopfoberfläche messbar sind. Die Spannungsschwankungen entstehen durch die Addition, der von einzelnen Neuronen erzeugten Ströme und lassen sich dadurch auch messen. Um die gemessenen Spannungsschwankungen auch klinisch bewerten zu können wird eine Aufzeichnung von verschiedenen Elektrodenkombinationen benötigt. Die erhaltenen Daten werden dann von Spezialisten oder einer Software zur automatischen Signalanalyse auf auffällige Muster untersucht. [22]

Die zu messenden Signale liegen in der Größenordnung von 5 bis 150 µV (1 Mikrovolt = 1 Millionstel Volt), aus diesem Grund ist der Einsatz eines empfindlichen Messverstärkers sinnvoll. Um das Netzbrummen und all die anderen Störungen zu unterdrücken, wird eine Differenzverstärker - elektrischer Verstärker, der Differenz zwischen zwei Eingangssignalen misst - mit hoher Gleichtaktunterdrückung eingesetzt. [22]

Um die Signale auch lesen zu können, werden Elektrodenkombinationen benötigt. Die Elektroden für das EEG werden in einem bestimmten System angebracht, dem 10-20-System. [22]

Beim 10-20-System handelt es sich um ein System zur Anordnung der Elektroden. Der Schädelknochen, wird vom Nasion bis zum Inion vermessen, und der daraus resultierende Wert als 100 Prozent angenommen. Danach wird dieser ausgehend vom Nasion mit 10 Prozent, vier mal 20 Prozent und nochmals 10 Prozent „abgesteckt". Anhand dieser Koordinaten werden danach die Elektroden auf der Kopfhaut angebracht. [23]

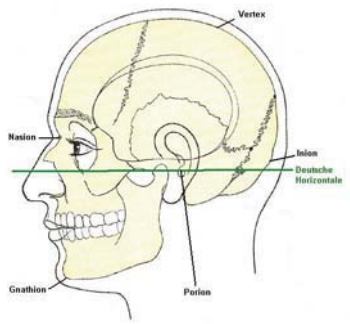


Abbildung 7: Schädelknochen – EEG Koordinaten [23]

Die daraus resultierenden Signale werden digitalisiert und auf einer Festplatte oder einem optischen Medium gespeichert und das EEG vom Neurologen am Bildschirm ausgewertet. Früher, vor der Computerisierung, wurden die Daten direkt zu einem Messschreiber geleitet, welcher das EEG auf Endlospapier schrieb, dabei wurden ca. 120 Blätter innerhalb einer Standarduntersuchung beschrieben. [22]

NICHT-INVASIVE ANWENDUNG

Um einen Einblick in die Anwendung der einzelnen nicht-invasiven Methoden zu bekommen, wird im folgenden auf drei konkreten Projekte eingegangen.

P300 – Prototyp eines BCI-Rollstuhls

Die Forscher Rebsamen, Teo, Zeng und Ang Jr. von der Universität Singapur haben in Kooperation mit einem Mitarbeiter des Londoner Imperial Colleges und Mitarbeitern des „ Institutes for Infocomm Research" einen Prototyp eines BCI gesteuerten Rollstuhles entwickelt. Dieser Rollstuhl verwendet ein P300 EEG Signal und eine Steuerungsstrategie, um innerhalb eines Gebäudes ohne Komplexe Sensoren oder Sensorverarbeitung zu navigieren. Die Steuerung des Rollstuhles beruht auf einem langsamen aber sicheren und genauen P300 EEG BCI, das den User den Bestimmungsort am Menü auswählen lässt. [13]

Abbildung 8: Rollstuhl mit EEG-Kappe und Computer [13]

Der Rollstuhl bewegt sich dann auf einem vorbestimmten Weg dem ausgewählten Ziel entgegen. Die Wege sind von einer Software definierte und nicht festverdrahtet, somit können sie leicht geändert werden, wenn sich das Umfeld ändert (siehe Abbildung 9). [13]

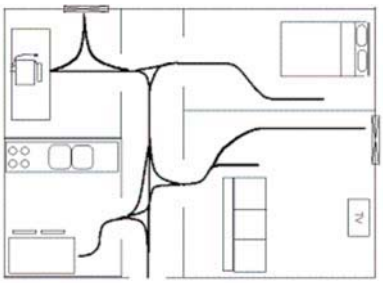


Abbildung 9: „Wegeplan" des Rollstuhles [25]

Roboter – Honda

Das japanische Unternehmen Honda Motor Corp. hat in Kooperation mit dem ATR Computational Neuroscience Laboratories die Verwendung von Gehirnsignale zur Steuerung einfacher Bewegungen eines Roboters erforscht.

Dabei wurden Forschungsteilnehmer in einem MRI Scanner positioniert. Der MRI Scanner erfasste die Gehirnaktivitäten und die dazu gehörigen Aktivitäten, d.h. wenn der Forschungsteilnehmer die Hand zu einer Faust machte, zeichnete der Scanner die dazu gehörigen Gehirnströme sowie das Ergebnis, also die Faust, auf. Diese Informationen werden gespeichert und wenn das System auf die Gehirnströme trifft, weiß es welche Aktion daraus folgen muss. Dies geschieht folgender Maßen: Die MRI Signale werden über Ethernet-Kabel, via TCP/IP zu einem Computer gesandt. Dieser verarbeitet die Informationen und verwendet dann die Software, um die Roboterhand die Befehle ausführen zu lassen. [10]

Abbildung 10: Honda Roboterhand [10]

Honda hat sich zum Ziel gemacht, in den folgenden zehn Jahren den gehenden Roboter „Asimo“ mittels dieser MRI Technologie weiterzuentwickeln. [10]

Passwort - Carleton

An der kanadischen Universität Carleton wurde ein Konzept für ein BCI System entwickelt, das als Basis für eine biometrische Identitätsstauthentifizierung dienen soll. Es gelang hier die traditionellen biometrischen Methoden, wie Fingerabdruckscanner oder Netzhautmusterscanner zu ersetzen.[10] Das Ziel dieses Systems, welches als Passwort-Gedanken Systems bezeichnet wird, ist so viel wie möglich an Entropie, also an Informationsgehalt, von den Gehirnsignalen eines Benutzers beim Übertragen seiner Gedanken, zu erhalten. [25]

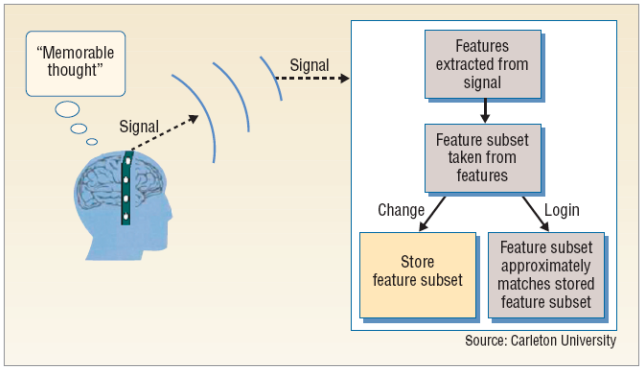


Abbildung 11: BCI Carleton - Funktionsweise [10]

In Abbildung 11 bekommt man einen guten Überblick über die Funktionsweise des Passwort-Gedanken Systems. Um diese auch wirklich verstehen zu können ist es von Vorteil den genauen Verwendungsablauf des Systems zu kennen[25]:

1. Wenn der User bereit ist betätigt er einen Button - denkt an das zuvor gewählte Passwort – und betätigt den Button erneut.
2. Elektroden notieren das Signal S, das während der Zeit zwischen dem ersten Betätigen des Buttons und dem zweiten Betätigen des Buttons ausgestrahlt wird.
3. S wird in die Signaleigenschaften F übersetzt.
4. Die Teilmenge der Eigenschaften $R \in F$ ist die, die den Gedanken des Users entspricht und über eine bestimmte Zeitdauer gleich bleibt. (R=Feature Subset)
5. R dient als eine Schablone, welche mit störungstoleranter Verschlüsselung gespeichert wird.
6. Das gespeicherte R wird zur Benutzerauthentisierung bei einem Computing-Device benutzt.

RESÜMEE/AUSBLICK

Das BCI ist eines der interessantesten aktuellen Forschungsgebiete im Bereich der Mensch-Maschine Interaktion. Mit dem Einsatz des BCI im alltäglichen Leben würde es zur Lösung sehr vieler Probleme kommen.

Vor allem für Menschen mit Behinderungen würde der Einsatz von BCIs sehr viele Erleichterungen mit sich bringen. Ein wesentlicher Vorteil wäre zum Beispiel die Steuerung von Prothesen mittels BCI. An der Universität in Stanford forschen Wissenschaftler bereits seit einiger Zeit nach Möglichkeiten die Signale zu identifizieren, die das Gehirn produziert, wenn es plant den Körper oder einen Teil davon in Bewegung zu versetzen. [10] Diese Art der Forschung hat eine sehr große Bedeutung, denn wenn es gelingt die Planungsaktivitäten genauestens zu erforschen, steht wirkungsvollen Prothesen nicht mehr im Wege. [10] Dies wäre einer der vielen Wege behinderten Menschen das Leben zu erleichtern. Ein weiterer Weg wäre, wie schon zuvor beschrieben, die Steuerung von Rollstühlen, Fernsehgeräten, Computer oder Licht mittels Gedanken. Aber auch die Steuerung bzw. Verwendung von Buchstabiermaschinen, wodurch stark behinderte Menschen (z.B. Lock-in Syndrom) wieder die Möglichkeit bekämen mit ihrer Außenwelt zu kommunizieren, wären durch den Einsatz vom BCIs möglich. [14]

Aber nicht nur für Menschen mit Behinderungen lösen sich Probleme durch die Verwendung von BCIs. Auch Menschen ohne Behinderung würden vom Einsatz der BCIs im realen Leben profitieren. Das Vergessen von Passwörtern würde zum Beispiel der Vergangenheit angehören, oder man könnte Computerspiele mittels Gedanken steuern. Wie bereits aus Science Fiction Filmen

bekannt, wäre auch das Steuern von Stereoanlagen, allein durch den Gedanken, möglich.

Doch ob diese Art von Technologie in Zukunft auch im realen Leben Anwendung finden wird, hängt von einer Vielzahl von medizinischen, ethischen und technologischen Faktoren ab. Wenn es gelingt, diese Faktoren auszuschalten, könnte das BCI schon bald auch im realen Leben eine wichtige und nicht mehr wegzudenkende Rolle spielen. Doch bis dahin bleibt uns nur eines: ABZUWARTEN und dem Beginn eines neuen Zeitalter entgegenzusehen - der Mensch-Maschinen Epoche.

REFERENZEN

[1] Cyberkinetics: BrainGate- Neural Interface System, URL: http://www.cyberkineticsinc.com/content/ medicalproducts/braingate.jsp, Downloaddatum: 6.6.2007

[2] T.Felzer: Verwendung verschiedener Biosignale zur Bedienung computergesteuerter Systeme, Universität Darmstadt, 2002.

[3] Konferenzpapier, EEG Konferenz 2000

[4] Lee U., Han, S.H., Kim H.S., Kim Y.B., Jung H.G., Lee H., Lang Y., Kim D., Jin M., Song J., Song S., Song C.G., Shin H.C.: Development of a Neuron Based Internet Game Driven by a Brain-Computer Interface System. In: *Proceeding of the International Conference on Hybrid Information Technology* – Volume: l2 (ICHIT'06) page 600-604, 2006.

[5] Hinterberger T., Baier G.: Parametric Orchestral Sonification of EEG in Real Time. In: *IEEE MultiMedia*, Volume: 12, Number 2, page 70-79, April/June 2005.

[6] Krepki R., Blankertz B., Curio G., Müller K.-R.: The Berlin Brain-Computer Interface (BBCI) - towards a new communication channel for online control in gaming applications. In: *Journal of Multimedia Tools and Applications*, 2007. (In print, online available: URL:http://www.springerlink.com/ content/l623732866024h72/)

[7] Millán J.: Brain-computer interfaces. In M.A. Arbib (ed.), "Handbook of Brain Theory and Neural Networks, 2nd ed." Cambridge: MIT Press, 2002.

[8] Nakatsu R., Tosa N.: Toward the Realization of Interactive Movies - Inter Communication Theater: Concept and System. In: *ACM, 1997 International Conference on Multimedia Computing and Systems (ICMCS'97)*, page 71, June 1997.

[9] Navarro K.-F.: Wearable, Wireless Brain Computer Interfaces In Augmented Reality Environments. In: *International Conference on Information Technology: Coding and Computing (ITCC'04)* Volume: 2, page 643, 2004.

[10] Ortiz S.: Brain-Computer Interfaces: Where Human and Machine Meet. In*: Computer Jannuary 2007*, Volume: 40, Issue: 1, page 17-21, 2007.

[11] Wikipedia: Positronen-Emissions-Tomographie, URL: http://en.wikipedia.org/wiki/ Positron_ emission_tomography, Downloaddatum: 30.5.2007

[12] Rebsamen B. et al: Controlling a Wheelchair Indoors Using Thought, In: *IEEE Computer Society Inteligent Systems*, Volume: 22, Number: 2, page:18-24, March/April 2007.

[13] Rebsamen B., Burdet E., Guan C., Zhang H., Teo C.L., Zeng Q., Laugier C., Ang M.H.Jr.: Controlling a Wheelchair Indoors Using Thought. In: *IEEE Computer Society*, Volume 22, Number 2, page 18-24, March/April 2007.

[14] J.Schönfeld, M.Gafert: Steuerung von Prothesen, Freie Universität Berlin, Institut Informatik, 2005

[15] J.Thorpe, P.C. van Oorschot, A.Somayaji: Pass-thoughts: Authenticating With Our Minds, IN: *ACM*, 2005

[16] S. Weber: Systeme für die Rehabilitation, URL: http://www2.mimed.mw.tum.de/Lehre/MHC/VL3 _070522.pdf, Downloaddatum: 8.6.2007

[17] Wikipedia: Elektrokortikogramm, URL: http://de.wikipedia.org/wiki/Elektrokortikogramm, Downloaddatum: 18.5.2007

[18] Wikipedia: Positronen-Emissions-Tomographie, URL: http://de.wikipedia.org/wiki/Positronen-Emissions-Tomographie, Downloaddatum: 7.6.2007

[19] Wikipedia: fMRI – funktionelle Magnetresonanztomographie, URL: http://de.wikipedia.org/wiki/FMRI, Downloaddatum: 10.6.2007

[20] Wikipedia: Magnetoenzephalographie, URL: http://de.wikipedia.org/wiki/Magnetoenzephalographie, Downloaddatum: 5.6.2007

[21] Wikipedia: Telsa, URL: http://de.wikipedia.org/wiki/Tesla_%28Einheit%29, Downloaddatum: 11.6.2007

[22], Wikipedia: Elektroenzephalographie, URL: http://de.wikipedia.org/wiki/Elektroenzephalografie

[23] Wikipedia: 10-20-System URL: http://de.wikipedia.org/wiki/10-20-System, Downloaddatum: 11.6.2007

[24] Wikipedia: SQUID, URL: http://de.wikipedia.org/wiki/SQUID, Downloaddatum: 12.6.2007

[25] NUS – National University of Singapore: Brain Controlled Wheelchair Project, URL: http://guppy.mpe. nus.edu.sg/~rebsamen/project/ Downloaddatum: 12.6.2007

Tangible Interfaces und Sensor-Network-Interface-Entwicklungen für den effizienten Datenzugriff

Martin Florian
Alpen-Adria Universität Klagenfurt
Universitätsstrasse 65-67, 9020 Klagenfurt
mflorian@edu.uni-klu.ac.at
+43 650 66 44 00 2

ABSTRACT
Der vorliegende Artikel beschäftigt sich mit der Speicherung, dem Zugriff, der Verwaltung, Organisation und Manipulation von digitaler Information. Vielerseits werden die in aktuellen Betriebssystemen zum Einsatz kommenden Ordner und Dateistrukturen als uneffizient und kompliziert angesehen [1][2]. Neue Interaktions-schnittstellen und technische Entwicklungen sollen den Umgang mit Computersystemen intuitiver und natürlicher gestalten. Im Fokus dieser Arbeit stehen Forschungsprojekte, deren Zielsetzung vor allem die wesentliche Erleichterung des Zugriffs auf digitale Information ist. Dabei werden im Wesentlichen zwei unterschiedliche Ansätze beschrieben, gegenübergestellt, deren Anwendbarkeit kritisch betrachtet und Möglichkeiten und Chancen eines gemeinsamen Einsatzes geprüft.

Author Keywords
Tangible Interface, Sensor Network Interface, manipulation of digital data, filesystem, desktop, Memodules, Siftables.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

EINLEITUNG

Die Interaktion mit Computern und damit die Manipulation digitaler Information über physische Gegenstände der realen Welt, so genannte `Tangible User Interfaces (TUIs)`, ist aktuell ein in der wissenschaftlichen Literatur umfangreich behandeltes Thema. Physische Objekte repräsentieren digitale Information und stellen so Mechanismen zur Einflussnahme auf digitale Systeme dar. Im Verlauf dieser Arbeit sollen dabei speziell Systeme behandelt werden, die es erlauben, vom Desktop als Metapher wieder zurück zum realen Schreibtisch zu kommen. Dies soll über angreifbare, sortier-, gruppier- und leicht wieder erkennbare, reale Objekte erfolgen. Anhand dieser sollen digitale Informationen direkt physisch manipulierbar sein. Der Benutzer kann - ohne Indirektion über abstrakte graphische Interfaces - natürliche und intuitive Ordnungsschemen erstellen und verwalten, was Effizienz und Benutzungsfreundlichkeit erheblich steigern soll [5][7].

DESKTOPMETAPHER

Der Fokus dieser Arbeit soll auf dem Handling, also dem Ablegen und Wiederfinden von digitalen Objekten liegen. Aktuell sind Computerbenutzer an verzweigte File-Systeme gebunden oder platzieren ihre aktuellen Daten auf dem Desktop, der jedoch als Metapher für den reellen Schreibtisch nicht überzeugen kann. Empirische Forschungsergebnisse belegen, dass vor allem Experten im Umgang mit Computern den Desktop seiner Intention nach verstehen, allerdings über die Hälfte der Standardbenutzer diese `Schreibtisch-Eigenschaft` nicht wahrnehmen oder nutzen. Aus entsprechenden Befragungs-Ergebnissen ist ableitbar, dass aber jene, die das Konzept wahrnehmen, entgegen denen die den Desktop rein als Hintergrundbild verstehen, dieses auch bewusst nutzen, um Effizienzvorteile zu erzielen und häufig genutzte Dateien und Verknüpfungen zum schnellen Zugriff am Desktop zu platzieren [4].

Die Desktopmetapher, wie sie in aktuellen Betriebssystemen Verwendung findet, ist allerdings bei weitem keine ausgereifte Lösung, da wesentliche Eigenschaften des realen Schreibtisches nicht abgebildet werden. Vor allem Ordnungsstrukturen sind in der zweidimensionalen Umgebung für den Bennutzer schwer einrichtbar. Dem entgegen wirkt die neue Entwicklung des `BumpTop` [1], einer 2.5D-Darstellung des Desktop mit wesentlich erweiterter Funktionalität, insbesonders hinsichtlich des Stapelns von Icons. 2.5D bezieht sich dabei darauf, dass die zweidimensionale Darstellung des Desktops geneigt wurde um über 3D-typische Merkmale, wie Schatten und Überdeckungseffekte, Stapel optisch leichter von einfachen Icons unterscheidbar zu machen.

Die Zwischenablage von digitaler Information in Form von Stapeln erweist sich als natürlichere und schneller

zugreifbare Form der Verwaltung als die Verwendung von Ordnern [13]. Auch was das Wiederfinden und die Suche in solchen, wenn auch digitalisierten Stapelsystemen, wie in Abb. 1 dargestellt, anbelangt ist eine beachtliche Effizienz zu erkennen.

Abb. 1. (a) Gewöhnlicher virtueller Desktop. (b) Ordnungsstrukturen auf einem realen Schreibtisch. (c) Darstellung des BumpTop mit Stapeln und Unterstüzung natürlicher Organisationsstrukturen [1]

Diese Effizienz wird durch empirische Untersuchungen deutlich, die zeigen, das selbst nach längerem zeitlichen Abstand zur letzten Verwendung, durchaus auch nach mehreren Monaten, die Zugriffszeit nicht signifikant höher ist [9], wohingegen der Zeitaufwand für das Durchsuchen von Ordnerstrukturen bis zum Wiederfinden des gewünschten Objekts mit steigendem zeitlichem Abstand deutlich wächst. Daher wurde in der Entwicklung des `BumpTop` besonderer Wert auf die dementsprechende Behandlung der Objekte und Stapel gelegt. Es wurden physikalische Eigenschaften von Objekten miteinbezogen um deren Manipulation so natürlich wie möglich zu gestalten, und Interaktionstechniken eingeführt, die das Gruppieren, Verschieben und Stapeln unterstützten [1]. Trotzdem bleibt das `BumpTop` Projekt vielleicht kein klassisches, aber immerhin ein durch die Darstellung und Indirektion der Eingabemöglichkeiten über Maus oder Keyboard zumindest teilweise beschränktes Graphisches User Interface.

Um diesen Einschränkungen entgegenzuwirken wurden in den letzten zehn Jahren im Rahmen folgend beschriebener Projekte daran gearbeitet digitale Information abseits des Bildschirms, in der realen Welt zu repräsentieren und so manipulierbar zu machen.

ERSTE ÜBERLEGUNGEN ZUR REPRÄSENTATION DIGITALER INFORMATION ÜBER PHYSISCHE OBJEKTE

Grundlegende Überlegungen die Abstraktion digitaler Ordnerstrukturen und File-Systeme umzukehren wurden 1998 im IBM Almaden Research Center angestellt. Speicherung und Wiederfinden von digitaler Information in verschachtelten Ordnerstrukturen und in komplexen Netzwerken wurde als aufwändig und zeitraubend empfunden. Für die Autoren war die Floppy-Disk das Mittel zum effizienten Datenaustausch, da sowohl lesender wie auch schreibender Zugriff auf die darauf gespeicherten Files ohne lange Navigationspfade möglich und die physische Ablage der Disk einfach handhabbar ist [2]. Problematisch erwiesen sich jedoch die mehr als eingeschränkte Speicherkapazität des Mediums und die damals fehlenden oder vergleichsweise teure Alternativen wie USB-Speichermedien.

Ebenfalls schränkt die Speicherung auf lokalen Medien die Kollaborationsmöglichkeiten ein, da immer nur der Benutzer der das Speichermedium bei sich hat auf die Daten zugreifen kann. Daraus wurde die Idee eines Pointer-Systems geboren, bei dem Repräsentation und tatsächliche Speicherung der Daten zur einfacheren Handhabung getrennt wurden. Forscher, die sich damit auseinander setzten, digitale Information außerhalb von Computersystemen zu repräsentieren, hielten an der bekannten Diskette fest, allerdings sollte dabei nicht die Information auf der Diskette gespeichert werden, sondern diese rein als physische Repräsentation des Zugriffspfads für den indirekten Zugriff auf die im Netzwerk gespeicherten Daten dienen. Jeder `Token`, bzw. jedes physische Objekt, welches digitale Information repräsentieren soll, würde eine eindeutige ID in einem Speichersystem haben. Das "Kopieren" von Daten auf dieses Objekt würde damit also eine Verknüpfung der Daten mit der ID des Objektes bedeuten, auf die dann darüber zugegriffen werden kann [2]. Die Datei kann somit, ohne sich um ein Ordnungsschema kümmern zu müssen, automatisch irgendwo abgelegt werden, da der Zugriff wiederum, ohne dass der Benutzer den genauen Speicherort kennen muss, durch den `Token` angestoßen wird.

Auch andere Überlegungen, abseits der Diskette, wurden in Betracht gezogen, wie Barcode-Systeme oder Infrarot-Transmitter, wodurch auch kabellose Technologien ins Spiel gebracht wurden [2].

AKTUELLE ENTWICKLUNGEN IM BEREICH TANGIBLER INTERFACES

Im Sinne des `Pervasive- oder Ubiquitous Computings` beschäftigt sich eine Forschungsgruppe des `MIT Media Laboratory` aktuell mit der Entwicklung eines völlig neuen User Interface Prototypen. Dieser soll der Interaktion mit Computersystemen intuitiver gestalten, die Aufgabenbewältigung unterstützen und dabei die Hardware, entsprechend den Ideen Weisers [12], in den Hintergrund treten lassen soll [7]. Die allgemein als Grundlage des allgegenwärtigen Einsatzes von Computersystemen in Alltag angesehene Vision Weisers [12], des integrierten und unsichtbaren Informations- und Kommunikatiossystems sah schon die Verwendung so genannter Pads, drahtlos vernetzter `scrap computers` vor, die in großer Anzahl zur Verfügung stehen. In seinem

Artikel von 1991 wurde eine wesentliche Eigenschaft der jetzt vorgestellten Forschungsarbeit vorweg genommen: „*Pads use the real desk, just as you spread out papers.*“ [12] Damit war gemeint, dass eben solche Repräsentationen von Daten die Beschränkung der Eingabemöglichkeiten in Computersyteme aufheben sollen und werden, und natürliche Arbeitsvorgänge, wie die Bearbeitung von Papierdokumenten, auch auf digitale Information angewendet werden kann.

Im Folgenden sollen zwei Projekte vorgestellt werden, die auf Basis vollkommen unterschiedlicher Technologien dieses Ziel weiter verfolgen. Während im einen Fall `Siftables` als so genannte generische Container zum Einsatz kommen, versucht das zweite Projekt mit seinem Ansatz der `Memodules`, implizite Assoziationen digitaler Information mit physischen Objekten zum effizienten Wiederfinden zu nutzen.

„Siftables“ - Generische Container

Augenscheinlich dieser Vision folgend kommen die Nachfahren der von Weiser beschriebenen `Pads` in Form von `Token` mit LCD-Displays und umfangreicher Sensor- und Netzwerktechnologie zum Einsatz. Die Zielsetzung des `Siftables` Projektes lautet eben den angesprochenen, durch die I/O-Möglichkeiten rein graphischer Benutzerschnittstellen bedingten Einschränkungen der Manipulation von digitalen Objekten entgegenzuwirken und so die Interaktion mit Computersystemen gemäß den Ideen Weisers in den Fokus zu stellen.

Die Zielsetzung des Projektes lautet, die Fähigkeit des Menschen zu nutzen, kleine physische Objekte schnell und effizient zu sortieren, manipulieren und dabei den Überblick zu behalten [7]. Daraus leitet sich auch das Homonym als Name des Projektes ab. So kann der Begriff aus der Bedeutung des englischen Wortes `sift` also `durchsuchen` oder `physisch wühlen` verstanden werden, andererseits ist darin das Acronym `SIFT` für `Sensor InterFace Token` zu finden.

Solche `Siftables`, wie in Abb. 2 dargestellt, sind so genannte `Generische Container`. Sie sind dazu angedacht, digitale Objekte in der realen Umwelt zu repräsentieren und die Möglichkeit zu bieten, diese Daten ohne weitere Computersysteme selbstständig zu manipulieren und Eingabeoperationen darauf auszuführen. Generisch bedeutet in diesem Zusammenhang, dass, im Gegensatz zu den später behandelten `Memodules`, jeder Typ von digitaler Information sinnvoll repräsentiert werden kann [5].

Dabei müssen die `Token` nicht selbst als Träger der digitalen Information, sondern nur als `Pointer` auf die tatsächliche Datei oder Anwendung dienen. So soll vor allem die Assoziationsfähigkeit des Menschen über, in der Literatur in Anlehnung an die aus klassischen GUIs bekannten Icons, also physische Icons bezeichneten, realen Abbildungen von Bits und Bytes, angesprochen werden [5].

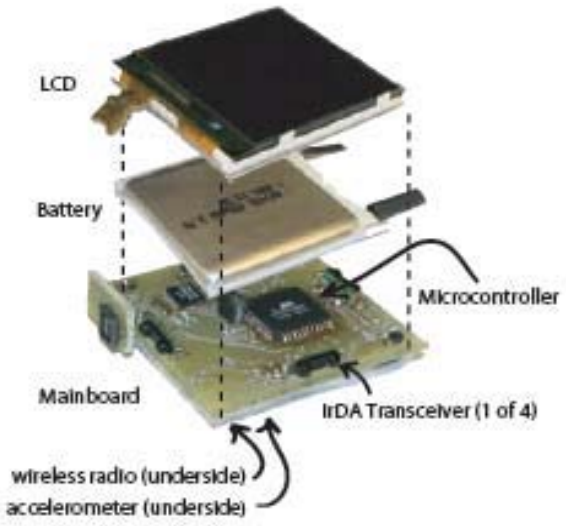


Abb. 2. Explosionsdarstellung der Komponenten eines Siftables [7]

Die Explosionsdarstellung eines solchen `Siftables` *zeigt die wesentlichen Hardware-Komponenten, die die besonderen Möglichkeiten der Datenmanipulation erlauben. Neben dem LCD-Display zur dynamischen Visualisierung der assoziierten Information, sind die in der Hauptplatine integrierten Kommunikationskomponenten sowie die Beschleunigungssensoren an der Unterseite die wesentlichen Komponenten die dieses neue Interaktionsmodell ermöglichen.*

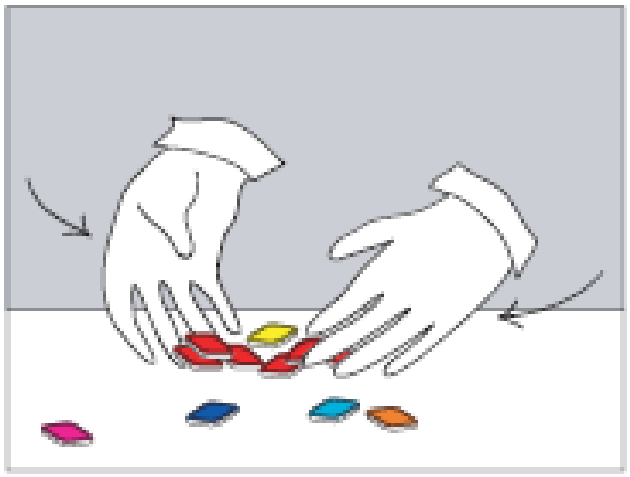


Abb. 3. Manipulation von Siftables [7]

`Siftables` *können einfach mit beiden Händen und unter Verwendung aller zehn Finger wie reale Dokumente am Schreibtisch angeordnet, sortiert und gestapelt werden, ohne Einschränkungen durch Unzulänglichkeiten der Eingabemethoden.*

Durch ihre Ausstattung mit drahtloser Netzwerktechnologie, Abstands- und Bewegungssensoren können diese physischen Icons durch Gestik als Eingabemedium genutzt oder räumlich am Schreibtisch angeordnet werden, wobei die so entstehenden Assoziationen und Kontextrelationen zwischen den Objekten automatisch auf deren digitale Entsprechung umgesetzt werden. Das Mapping digitaler Information auf physische Token kann somit eine wesentliche Vereinfachung der Arbeit darstellen, da dadurch, wie Abb. 3. zu veranschaulichen versucht, die menschliche Fähigkeit, physische Objekte effizient und einfach zu sortieren und zu organisieren, gefördert und genutzt wird [7].

Die Organisation physischer Objekte auf dem Schreibtisch folgt einem natürlichen intuitiven Muster. Dabei werden meist Stapel gebildet, in denen Informationen in weiterer Folge nach Wichtigkeit oder einfach chronologisch aufeinander gelegt und gruppiert werden [6].

Siftables vermögen die unterschiedlichsten Daten zu repräsentieren, erlauben aber auch deren direkte Manipulation. Auf Grund dieser beiden Eigenschaften ist eine eindeutige Kategorisierung, wie von Holmquist [5] vorgesehen, nicht ohne weiteres möglich. Am ehesten zutreffend bezeichnen könnte man die von ihrer Funktion her mächtigen, anfassbaren Interaktionsmechanismen daher als hybride Mischform zwischen generischem Container und Tool. Die Einordnung als generischer Container ist zutreffend, da die Siftables durch ihre LCD-Displays dynamisch sind und unterschiedlichste Daten aufnehmen und darstellen können. Ihre umfangreiche technologische Ausstattung erlaubt es aber, auch direkte Eingaben zu machen und im Sinne eines eigenständig berechnenden Computers eingesetzt zu werden, was ebenfalls eine Bezeichnung als Tool erlaubt [5].

Im Bereich Sortierung und Gruppierung sowie Systemeingaben, welche im Fokus der Entwicklung lagen, ist dieses Projekt nach ausgiebigem Literaturstudium sicherlich als die aktuelle Speerspitze auf dem Forschungsgebiet physischer Objekte zu Repräsentation und Manipulation von Daten zu bezeichnen, allerdings gibt es in diesem und ähnlichen Forschungsgebieten zahlreiche Entwicklungen die andere, wie anhand der Memodules dargestellt, wesentliche Eigenschaften und Vorzüge aufweisen.

Die Siftables sind ohne Zweifel eine technisch herausragende Lösung, um digitale Information in der realen Welt anfassbar und manipulierbar zu machen. Mit der sich noch verstärkenden digitalen Informationsflut zu Recht zu kommen, wird aber zunehmend schwieriger: Zielgerichtet nach Dateinamen oder Icons zu suchen endet meist in Resignation. Dementsprechend verhält es sich auch mit generischen Containern, die einerseits zwar beliebige Information repräsentieren können, wobei anderseits aber die Unterscheidbarkeit ab einer gewissen Anzahl nicht mehr gegeben ist und dadurch das Konzept deutlich an praktischer Bedeutung verliert.

Anders verhält sich dies jedoch bei natürlichen Objekten in der realen Welt, die anhand ihrer physischen Eigenschaft wesentlich besser wahrgenommen und unterschieden werden können [8].

EXKURS RFID-TECHNOLOGIE

Zum Verständnis des folgend vorgestellten Projektes ist grundlegendes technisches Wissen über die Funktionsweise der RFID-Technologie notwendig, welches an dieser Stelle vermittelt werden soll.

Der Datentransfer bei diesen passiven RFID-Systemen, also solchen, die mit miniaturisierten, ohne eigene Stromversorgung auskommenden Funkchips arbeiten, basiert auf dem Prinzip der Induktion, was bedeutet, dass die Energie für die Kommunikation aus einem vom Reader abgestrahlten magnetischen Feld kommt, welches vom RFID-Label dazu verwendet wird, ein Rücksendesignal zu modulieren und die gespeicherten Daten auf diesem Weg zu übertragen [10].

Die Versorgung des Chips mit Strom beruht auf dem, von Michael Faraday bei Versuchen, die Funktionsweise eines Elektromagneten („Strom erzeugt Magnetfeld“) umzukehren („Magnetfeld erzeugt Strom“), erforschten physikalischen Prinzip, dass in einem Leiter (Draht) elektrische Spannung entsteht, wenn dieser von einem Magnetfeld "geschnitten" wird. Dies ist der Fall, wenn ein Leiter quer durch ein Magnetfeld bewegt wird, oder, wie bei passiven RFID-Systemen, sich um ihn herum ein Magnetfeld auf- oder abbaut. Die Reichweite hängt also von der magnetischen Feldstärke ab, die demnach groß genug sein muss, um den RFID-Tag zu aktivieren [3].

Zum allgemeinen Verständnis sei noch angemerkt, dass es dabei häufig vorkommt, dass sich zeitgleich mehrere RFID-Tags im Lesebereich des Readers befinden und somit alle aktiviert werden, dafür aber unterschiedlichste Anti-Kollisionsmechanismen bestehen die dieses Problem adressieren [10].

„Memodules“ - Assiziation digitaler Information mit physischen Objekten

In den letzten Jahrzehnten wurden mehrere, wenn auch technologisch deutlich weniger mächtige und herausragende, einfachere Systeme entwickelt, die einen großen Vorteil haben: Den assoziativen Wiedererkennungswert und die Darstellung physischer Eigenschaften der verknüpften Daten.

"Token-Based Access to Digital Information: A system where a physical object (token) is used to access some digital information that is stored outside the object, and where the pysical representation in some way refelects the nature of the digital information it is associated with."[5]

Das schweizerisch italienische Projekt Memodules [8] setzt wie ein Xerox PARC System [11] auf natürliche Objekte der realen Welt, an die über RFID-Technologie digitale Information geknüpft wird. Dabei wird die Assoziationsfähigkeit des Benutzers optimal unterstützt.

Die digitalen Urlaubsfotos des letzten Jahres im privaten Netzwerk wieder zu finden, kann schon zur zeitraubenden Aufgabe werden, noch dazu die elektronische Hotelbuchung wieder zu finden, scheint oft aussichtslos. Anders jedoch verläuft die Suche nach den Souvenirs vom Strand, deren Aufstellungsplatz im Wohnzimmer, oder auch deren Lagerplatz im Keller man noch deutlich vor Augen hat und deren charakteristische physische Eigenschaften gut im Gedächtnis verankert sind.

Diesen Umstand machen sich Forscher unterschiedlicher Einrichtungen zu nutze, um den Zugriff auf digitale Information zu erleichtern. Wie in Abb. 4 dargestellt, sollen prägnante physische Objekte mittels RFID-Technologie zu Pointern auf digitale Information werden. Solche Systeme sind jedoch passiv, was bedeutet, dass von den Token selbst keine Aktion ausgehen kann und sich ihre Funktion darauf beschränkt, über ihre ID einen Pointer auf verknüpfte Informationen darzustellen. Der ID des Funkchips werden digitale Informationen zugeordnet, die dann bei Verwendung des Tokens abgerufen werden können.

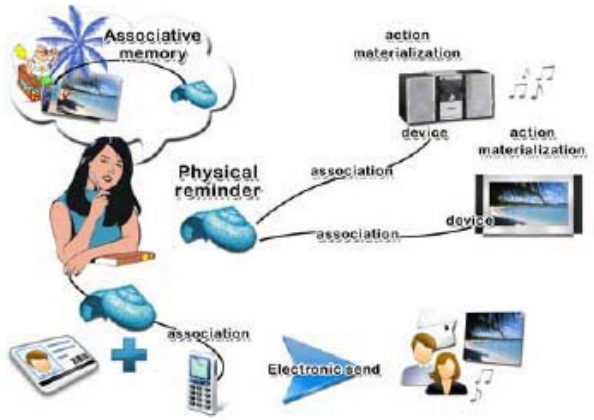


Abb. 4. Verwendung von Memodules als physische Token zum Zugriff auf assoziierte Daten und Anwendungen. [8]

Diese müssen jedoch immer wieder über "gewöhnliche" GUIs initialisiert werden. Vorteilhaft dabei ist sicherlich die Möglichkeit der chaotischen automatischen Speicherung der Information, da der Zugriff dann ohnehin über das Objekt erfolgt.

Contextawareness

Möglich ist auch ein overloading eines Tokens [5], was bedeutet, dass zu einem physischen Gegenstand bzw. zu dem dazugehörigen RFID-Tag mehrere digitale Informationen assoziiert werden. Darüber ist auch bis zu einem gewissen Grad contextawareness gegeben:

Je nachdem mit welchen anderen Token das Objekt im Verbund eingesetzt, das heißt, in den Lesebereich des RFID-Readers gebracht wird, werden unterschiedliche Assoziationen genutzt und dementsprechende Aktivitäten und Applikationen ausgeführt. Dadurch kann ein Token in unterschiedlichen Zusammenhängen unterschiedliche Informationen repräsentieren [5].

KOMBINATION VON SIFTABLES UND MEMODULES

Im Zusammenhang mit Contextawareness treten auch wieder die durch die umfangreiche technologische Ausstattung der Sensor-Network-Interfaces bedingten Vorteile der Technologie deutlich zu Tage. Bei genauerer Betrachtung kommt zum Vorschein, dass ihre wesentlichen Bestandteile darauf ausgerichtet sind, eigenständig und ohne weitere Applikation räumliche Relationen zu anderen Token oder Netzwerkkomponenten zu erfassen und so die kontextabhängige Verwendung ermöglichen.

So kann beispielsweise auch die Initialisierung der Siftables ohne andere Benutzerschnittstelle vollzogen werden, da es die Sensorik emöglicht, Assoziationen zwischen mehreren dieser Token beispielsweise durch "Zusammenschütteln" herzustellen und via Gestik Eingaben in das System zu machen [7].

Beide Technologien haben demnach einige sehr gute Ansätze, die den Umgang mit digitaler Information mit Sicherheit erleichtern können, nichts desto trotz bleiben jeweils wesentliche Probleme unbehandelt. Einerseits der Wiedererkennungswert, andererseits ist die Initialisierung und Verwendung mit Problemen behaftet. Einen Ausweg könnte die Kombination beziehungsweise der gemeinsame Einsatz der beiden Technologien bringen. Um das System optimal nutzen zu können, könnten die Siftables seitens der Hardware um RFID-Komponenten erweitert werden. Die Schwierigkeit dabei ist es wahrscheinlich nicht die Sensor-Network-Token um RFID-Tags zu erweitern, um aber wirklich einen Vorteil zu erzielen, müssten diese auch den umgekehrten Weg beschreiten und aktiv mit den Funkchips der Memodules umgehen können.

Das Problem, das sich dabei ergeben dürfte, die Siftables zu erweitern, ist also jenes, dass diese den RFID-Chip lesen und beschreiben können müssten, selbst aber drahtlos und abhängig von einer kleinen Batterie sind, wodurch es schwierig wird, ein solches elektromagnetisches Feld aufzubauen, um die direkte Kommunikation mit den Memodules zu ermöglichen.

Ebenso benötigt der Reader im Gegensatz zum Chip doch einiges an Platz, wodurch die Integration auf ein mobiles Gerät unwahrscheinlich erscheint. Kann für dieses Problem eine Lösung gefunden werden, scheint ein gemeinsamer Einsatz der beiden Forschungsergebnisse aber sehr lohnend zu sein.
Ein Ansatz, um die so entstehenden Möglichkeiten zumindest eingeschränkt nutzen zu können, wäre es, über die Indirektion einer Basisstation zu arbeiten, die dann zwar Mobilität und Flexibilität einschränken, aber zumindest eine Kommunikation zwischen den unterschiedlichen Token erlauben würde.

INITIALISIERUNG DER MEMODULES MITTELS ERWEITERTER SIFTABLES

Die Siftables können zur Strukturierung und Manipulation der Daten verwendet werden, um dann zur „Aufbewahrung" der Daten, respektive zur Langzeitspeicherung und um die Wiederauffindbarkeit zu fördern, ihre Assoziationen auf die „natürlichen", physischen Token übertragen und diese so ohne Verwendung eines herkömmlichen Computers oder anderer abstrakter Interfaces initialisieren. Eine ideale Lösung dafür wäre es, die mit dem RFID-Tag ausgestatteten Memodules einfach entsprechend nahe an den im Siftable integrierten Reader/Writer heranzuführen und sobald damit eine Datenverbindung zwischen den beiden besteht, die Assoziationen der Token durch gemeinsames Schütteln, welches von den Beschleunigungssensoren im Siftable als Input interpretiert werden kann, herzustellen.
Da dies jedoch, wie schon angesprochen, durch die technischen Probleme schwierig ist, besteht die Alternative nicht, die Siftables zu erweitern, sondern die Memodules-Konsole, die aktuell zur Initialisierung der RFID-Labels (mittels klassischen GUI) verwendet wird zu überarbeiten. Diese müsste um eine WLAN-Fähigkeit erweitet werden, um auch mit den Siftables kommunizieren zu können. In der Folge können die Assoziationen der Siftables dann über diese Konsole auf die „natürlichen" RFID-Token übertragen werden, wodurch ein solches kombiniertes System die Vorteile beider Technologien ausnützen und die jeweils eigenen Schwächen ausmerzen könnte. Der gemeinsame Einsatz erlaubt somit eine wirkliche Vereinfachung des Zugriffs auf Daten und Applikationen.

VOR-, NACHTEILE UND PROBLEME BEI DER VERWENDUNG VON PHYSISCHEN TOKEN

Die physische Anordnung der digitalen Daten im realen dreidimensionalen Raum unterstützt auf jeden Fall die Handhabung und Übersichtlichkeit und ist der Anordnung von Verknüpfungen am Desktop oder der Ablage in virtuellen Ordernstrukturen aus Sicht der Usabilty der Datenverwaltung am Arbeitsplatz überlegen. Anders sieht es jedoch aus, wenn man ein anderes Kriterium des Pervasive Computings zur Beurteilung heranzieht: die jederzeitige, aber vor allem mobile Verfügbarkeit der Daten.

Sämtliche der hier angeführten Technologien setzten darauf, die Daten im Netzwerk zu speichern, wodurch der Idee des Ubiquitous Computing grundsätzlich Rechnung getragen wird. Diese dann aber nur über die Token ansprechen zu können birgt wiederum Probleme. Die Frage, die sich dadurch ergibt, ist, in wie weit der Nutzer tatsächlich immer alle seine Token bei sich hat, um an die dazugehörige Informationen zu gelangen. Auch die generischen Container sind nicht dazu gedacht, alle Daten eines Benutzers zu repräsentieren. Ganz im Gegenteil, ihre Aufgabe ist es, einige wenige logisch zusammengehörige Informationen aufzunehmen und nicht wieder unübersichtlich große Datenmengen zu verkörpern. An dieser Stelle ergibt sich somit das Problem, dass ein Datenzugriff auf entsprechend freigegebenen Ressourcen, wie es momentan zwar vielleicht von vielen als umständlich und kompliziert empfunden wird, aber jederzeit möglich ist, nur mehr möglich sein sollte, wenn die Token physisch mitgetragen werden. Wie oft ergibt sich allerdings die Situation, dass spontan Daten benötigt werden, deren Suche dann zwar möglicherweise langwierig ist und wo der Zugriff über komplexe, problembehaftete und wenig benutzerorientierte Schnittstellen erfolgen muss, dies aber immerhin in den meisten Fällen in akzeptabler Zeit möglich ist? Auch wenn im Vorfeld bekannt ist, welche Daten benötigt werden, stellt sich die Frage, in wie weit es zielführend und praktisch ist, anstatt eines Laptops eine größere Anzahl solcher Token mit sich zu schleppen, nur weil deren Verwendung dann intuitiver und „natürlicher" sein soll.

Sicherlich bieten diese neuen Interaktionsmöglichkeiten am lokalen Schreibtisch einiges an Potential, sobald aber Mobilität gefordert ist, stoßen sie an ihre Grenzen. Eine duale Verwendung klassischer Desktop- und Ordnerstrukturen scheint aber auch als kritisch betrachtet werden zu müssen. Der große Vorteil der Token ist, dass diese selbst den Zugriffspfad auf die Datei repräsentieren und demnach der tatsächliche Speicherort der Daten unerheblich ist und die Verwaltung von Bits und Bytes in Speicherstrukturen chaotisch oder automatisch erfolgen kann, ohne dass dabei auf für den menschlichen Benutzer logische Pfade geachtet werden muss. Daraus folgt aber, dass ein alternativer Zugang zu den Daten über Ordnerstrukturen nicht mehr gegeben ist und die Token so die einzige effiziente Zugriffsmöglichkeit sind.

Die Alternative, Dateien nach eigenen Kriterien so in Ordnern abzulegen, dass sie auch über klassische, graphische Interfaces, zusätzlich zum Zugang über den Token, wieder gefunden werden können, bedeutet jedoch wieder den, als den Ursprung allen Übels und als Problemstellung aller diesbezüglicher Forschungsarbeiten definierten Aufwand über unzulängliche Interfaces abstrakte Ordnerstrukturen aufzubauen und zu verwalten.

RESÜMEE

`Tangible Interfaces`, weiterführende Entwicklungen und `Token` - in ihrer Intention, physische Pointer auf digitale Daten dazustellen, - bieten enorme Möglichkeiten Desktop- und Datenverwaltung zu vereinfachen, ihre Anwendung bringt aber auch Probleme. Bislang gibt es kein System, dass es erlaubt elektronische Dokumente wirklich unabhängig von klassischen Computern und deren graphischen Benutzerschnittstellen wie Artefakte der realen Welt zu benutzen. Die in diesem Artikel vorgeschlagene Kombination zweier Systeme könnte dem nahe kommen. Trotzdem bleiben einige gravierende Probleme, wie die angesprochene Mobilität, bestehen, die aktuelle Systeme besser lösen, wodurch die Benutzbarkeit und Akzeptanz schwerwiegend beeinträchtigt sein dürfte. Zukünftige Entwicklungen, welche diese Probleme gezielt adressieren und ausmerzen, haben aber mit Sicherheit großes Potential die Schnittstellengestaltung zwischen Computer und Mensch zu revolutionieren.

REFERENCES

1. Agarawala, A. and Balakrishnan, R. 2006. Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM Press, New York, NY, 1283-1292.
2. Barrett, R. and Maglio, P. P. 1998. Informative things: how to attach information to the real world. In *Proceedings of the 11th Annual ACM Symposium on User interface Software and Technology* (San Francisco, California, United States, November 01 - 04, 1998). UIST '98. ACM Press, New York, NY, 81-88.
3. Finkenzeller K., 1998. RFID-Handbuch – Grundlagen und praktische Anwendungen induktiver Funkanlagen, Transponder und kontaktloser Chipkarten, Hanser Verlag, München/Wien 1998.
4. Graf, M. 2007. Die Desktop-Metapher – eine empirische Studie und Implikationen, Diplomarbeit, Universität Klagenfurt.
5. Holmquist. L. E., Redstöm, J., Ljungstrand, P. 1999. Token-Based Acces to Digital Information, Viktoria Institute, Gothenburg/Sweden, URL (05/2007): http://www.viktoria.se/fal/publications/play/1999/token.pdf
6. Malone, T. W. 1983. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.* 1, 1 (Jan. 1983), 99-112.
7. Merrill, D., Kalanithi, J., and Maes, P. 2007. Siftables: towards sensor network user interfaces. In *Proceedings of the 1st international Conference on Tangible and Embedded interaction* (Baton Rouge, Louisiana, February 15 - 17, 2007). TEI '07. ACM Press, New York, NY, 75-78.
8. Mugellini, E., Rubegni, E., Gerardi, S., and Khaled, O. A. 2007. Using personal objects as tangible interfaces for memory recollection and sharing. In *Proceedings of the 1st international Conference on Tangible and Embedded interaction* (Baton Rouge, Louisiana, February 15 - 17, 2007). TEI '07. ACM Press, New York, NY, 231-238.
9. Robertson, G., Czerwinski, M., Larson, K., Robbins, d. Thiel, D., & van Dantzich, M. (1998). Data Mountain: Using spatial memory for document management, *UIST*. p. 153-162, *In: [1].*
10. Schoblick R., Schoblick G., 2005. RFID Radio Frequency Identification – Grundlagen, eingeführte Systeme, Einsatzbereiche, Datenschutz, Praktische Anwendungsbeispiele, Franzis Verlag, Poing 2005.
11. Want, R., Fishkin, K. P., Gujar, A., and Harrison, B. L. 1999. Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit* (Pittsburgh, Pennsylvania, United States, May 15 - 20, 1999). CHI '99. ACM Press, New York, NY, 370-377.
12. Weiser, M. 1999. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3, 3 (Jul. 1999), 3-11.
13. Whittaker, S. and Hirschberg, J. 2001. The character, value, and management of personal paper archives. *ACM Trans. Comput.-Hum. Interact.* 8, 2 (Jun. 2001), 150-170. *In: [1].*

FURTHER READING

Fertig, S., Freeman, E., and Gelernter, D. 1996. “Finding and reminding” reconsidered. SIGCHI Bull. 28, 1 (Jan. 1996), 66-69. DOI= http://doi.acm.org/10.1145/249170.249187

Karypidis, A. and Lalis, S. 2003. The Tangible File System. In Proceedings of the 23rd international Conference on Distributed Computing Systems (May 19 - 22, 2003). ICDCSW. IEEE Computer Society, Washington, DC, 268.

Barreau, D. and Nardi, B. A. 1995. Finding and reminding: file organization from the desktop. SIGCHI Bull. 27, 3 (Jul. 1995), 39-43.

Kindberg, T. 2002. Implementing physical hyperlinks using ubiquitous identifier resolution. In Proceedings of the 11th international Conference on World Wide Web (Honolulu, Hawaii, USA, May 07 - 11, 2002). WWW '02. ACM Press, New York, NY, 191-199.

Ljungstrand, P., Redström, J., and Holmquist, L. E. 2000. WebStickers: using physical tokens to access, manage and share bookmarks to the Web. In Proceedings of DARE 2000 on Designing Augmented Reality Environments (Elsinore, Denmark). DARE '00. ACM Press, New York, NY, 23-31.

Interaktive Oberflächen – Verschiedene Technologien im Vergleich

Ursula Dittrich
Universität Klagenfurt
udittric@edu.uni-klu.ac.at

ABSTRACT
Mit Interaktiven Oberflächen wird nicht wie gewohnt mit der Tastatur oder einer Maus interagiert. Genutzt werden bisher für die Arbeit am Computer ungenutzten Oberflächen, wie zum Beispiel eine Tischplatte. Mittels Gesten und Berührung wird mit dem Rechner kommuniziert. Interaktive Oberflächen werden häufig in Kombination mit Tangible User Interfaces (mit greifbaren Objekten Interagierende Benutzerschnittstellen) und "Hands-free" Computing eingesetzt. Der Benutzer muss keine speziellen Eingabegeräte bedienen um mit dem Rechner zu kommunizieren.

Das vorliegende Paper beschäftigt sich mit den physikalischen Grundlagen, der Zusammensetzung und dem Aufbau von feststehenden Interaktiven Oberflächen. Weiters werden die verschieden Technologien zur Umsetzung einer Interaktiven Oberfläche miteinander verglichen und deren Vor- und Nachteile hervorgehoben. Einen zusätzlichen Schwerpunkt des Papers stellen verschiedenste Kombinationsmöglichkeiten dieser Technologien dar.

In der Einführung wird der Leser zur Nutzung Interaktiver Oberflächen motiviert. Im Weiteren werden die Technologien erklärt und deren Anwendungsbereiche anhand von Beispielen vorgestellt. Besonderes Augenmerk wird der Interaktion mit Interaktiven Oberflächen und deren Integrierung in den Alltag zukommen. Die Arbeit schließt mit einem Ausblick auf mögliche, zukünftige Anwendungen.

Author Keywords
Interaktive Oberflächen, Interaktion, Tangible Interfaces, Hands-free Computing, SmartSkin, UbiTable, Smartboard, Augmented Reality,

EINFÜHRUNG

Interaktive Oberflächen sind ein Forschungsgebiet im Bereich der Erweiterten Realität (Augmented Reality). Das Ziel dieses Forschungsbereiches ist es bisher ungenutzte Oberflächen interaktiv zu machen und mit deren Hilfe mit der Umwelt zu interagieren. Mit Interaktiven Oberflächen wird nicht wie gewohnt mit der Tastatur oder einer Maus interagiert. Genutzt werden bisher für die Arbeit am Computer ungenutzten Oberflächen, wie zum Beispiel eine Tischplatte. Mittels Gesten und Berührung wird mit dem Rechner kommuniziert. Ein Beispiel hierfür sind interaktive Werbeflächen. Beispielsweise könnte die Schaufensterscheibe eines Autohauses interaktiv gestaltet werden. Auf der Scheibe könnte dann zum Beispiel eine Anwendung zur Fahrzeugkonfiguration laufen. Mittels Sensortechnik wäre es sogar möglich festzustellen, ob jemand vor der Scheibe steht und die potenziellen Kunden Kunde könnte direkt angesprochen werden (siehe Abbildung 2).

Abbildung1. Passanten bedienen eine Interaktive Oberfläche auf einem Schaufenster [1].

Ein typisches Charakteristikum der Interaktiven Oberfläche ist die Benutzung der Hand als Eingabemedium. Es gibt keine Maus und keine Tastatur.

BEWEGGRÜNDE FÜR DIE ENTWICKLUNG INTERAKTIVER OBERFLÄCHEN

In [3] wird als Motivation für die Entwicklung einer Interaktiven Oberfläche folgendes Szenario angegeben: Zwei oder mehrere Personen sitzen an einem Tisch, jeder hat sein Laptop als private Sicht vor sich. Jedes Mitglied der Gruppen sieht auf den eigenen Bildschirm. Andere Teilnehmer können eventuell noch Teile von diesem einsehen. Diese Arbeitssituation schließt gemeinsames Arbeiten z.B. an einem Vortrag aus.

Durch Verwendung einer Interaktiven Oberfläche kann dieses Problem gelöst werden. Das Projekt UbiTable unternimmt zum Beispiel den Versuch, einen interaktiven Tisch zu schaffen, an dem Personen ihre Daten flüssig austauschen und manipulieren können. Indem die Ein- und Ausgabe auf der Tischoberfläche stattfindet (siehe Abbildung 2) kann jeder Mitarbeiter am Arbeitsablauf aktiv

teilnehmen. Die Arbeitsoberfläche ist für alle Benutzer einsehbar. Der Tisch bietet USB-Schnittstellen, um Laptops, Kameras oder andere USB-Geräte schnell an das System anzuschließen. Ziel des Einsatzes eines UbiTables ist es Daten, die auf mitgebrachten Laptops gespeichert sind über ad-hoc Netzwerke[1] auf dem UbiTable darzustellen und in der Gruppe zu manipulieren.

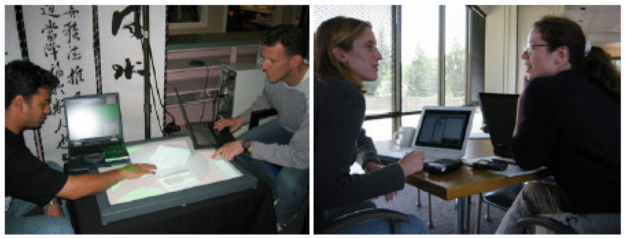


Abbildung 2. Vergleich zwischen der Zusammenarbeit am UbiTable und ohne derartige Hilfsmittel[3]

Die ursprüngliche Motivation zur Entwicklung der Interaktiven Oberflächen war dementsprechend die Unterstützung kollaborativer Arbeiten in Arbeitsgruppen. Das Einrichten einer Arbeitsgruppe hat zum Ziel ein Ergebnis zu erreichen. Dazu muss auf optimale Art und Weise miteinander kommuniziert werden können.

Interaktive Oberflächen können die Kommunikation innerhalb einer Gruppe wesentlich verbessern, da jeder Teilnehmer den Inhalt ändern kann und somit iterativ gemeinsam ein Ergebnis erarbeitet wird. Missverständnisse werden schneller offensichtlich, da durch Manipulation der Interaktiven Oberfläche für die anderen Teilnehmer schnell klar wird, ob man die Probleme und deren Lösung ähnlich sieht. Darüber hinaus ist bekannt, dass der Lerneffekt beim Involvieren der Lernenden im Vergleich zur visuellen und mündlichen Wissensvermittlung wesentlich größer ist. Das heißt Interaktive Oberflächen bieten auch erweiterte Möglichkeiten der Wissensvermittlung [2].

MERKMALE UND GENERELLER AUFBAU

Eine Interaktive Oberfläche setzt sich aus drei Elementen zusammen: Einer Fläche als Darstellungsebene, wie zum Beispiel ein Tisch, einem System, das die Position der Hand erkennt und einem Grafikgenerator. Die Darstellungsebene ist bei den in diesem Paper betrachteten Interaktiven Oberflächen fest. Neben der grafischen Darstellung, die meist durch einen Projektor oder einen Monitor realisiert wird, ist dabei vor allem eine effiziente und fehlerlose Identifizierung und Verfolgung der Hand bzw. der Hände und der Bewegungen des Benutzers wichtig. Hierfür wurden verschiedene Erkennungsmethoden entwickelt: optische Erkennung, Infraroterkennung, Erkennung mittels Ultraschall, Lasermessung, kapazitive Erkennung und die Erkennung mittels Schall. Diese Techniken werden im folgenden Kapitel genauer besprochen.

[1] Für Ad-hoc Netzwerke ist keine besondere Konfiguration der Software notwendig damit Geräte miteinander kommunizieren können. Bluetooth und ZeroConf für IP-Netze sind Beispiele hierfür.

TECHNIKEN ZUR ERKENNUNG DER HAND

Interaktive Oberflächen benutzen wie schon erwähnt zur Kommunikation mit dem Rechner weder Tastatur noch Computermaus. Vielfach wird auf der Interaktiven Oberfläche direkt mit der Hand oder den gesamten Arm manipuliert [6]. Die Erkennung der Position der Hand muss in Echtzeit erfolgen, dasselbe gilt auch, wenn man mit der Hand einen virtuellen Gegenstand auf der Interaktiven Oberfläche manipulieren will.

Bei der Umsetzung der Interaktiven Oberfläche sind folgende Grundsätze zu beachten:

- „Come as they are“: die Benutzern können direkt die Oberfläche manipulieren ohne sich vorkonfigurieren zu lassen oder zusätzliche Eingabengeräte verwenden zu müssen. Diese Eigenschaft sichert der Interaktiven Oberfläche die Flexibilität zu, dass sie von jedem Benutzer jeder Zeit ohne Vorbereitung manipuliert werden kann.
- Einfache und günstige Umsetzung: das System soll im Alltag eingesetzt werden können und jeder potentielle Benutzer soll es sich leisten können.
- Effizienz: da das zu entwickelnde System interaktiv sein soll, wird eine möglichst kleine Verzögerung erwartet.
- Hardwarerobustheit: die Hardware soll so robust gestaltet sein, dass das System, neben den zusätzlichen Eigenschaften, auch als normaler Alltagsgegenstand benutzt werden kann.
- Korrektheit: das zurück gelieferte Ergebnis auch unter kritischen Bedingungen zuverlässig sein.
- Rückstandstoleranz: die Gegenstände die sich auf der Oberfläche befinden, sollen die normalen Operationen nicht behindern bzw. stören.

Im folgenden Abschnitt werden die verschiedenen Erkennungsverfahren kurz erklärt und deren Vor- bzw. Nachteile dargelegt.

Optische Erkennung

Bei diesem Verfahren werden Kameras über der jeweiligen Fläche installiert. Diese nehmen ein oder mehrere Bilder gleichzeitig aus unterschiedlichen Perspektiven auf. Mittels Bildverarbeitung wird aus den jeweiligen Bildströmen die aktuelle Position der Hand ermittelt. Durch die Differenzbildung zwischen dem Bild der Projektion und den Aufnahmen der Kamera kann aus dem Kamerabild die exakte Positionierung der Hand herausgerechnet werden. Ein entscheidender Vorteil dieser Technik besteht darin, dass der Abstand zwischen Kamera und Hand immer

annähernd konstant bleibt und dadurch die Genauigkeit erhöht wird , die Verarbeitung beschleunigt und Fehlerreduziert werden können (siehe Abbildung 3).

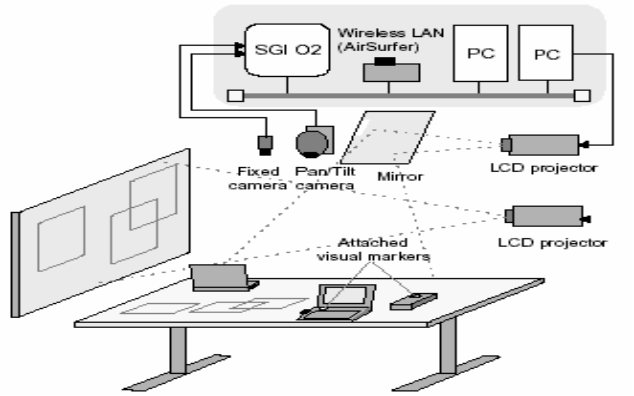


Abbildung 3. Aufbau einer Interaktiven Oberfläche, die auf optischer Erkennung basiert[8].

Ein Vorteil der Bildverarbeitung ist die Möglichkeit viele Objekte gleichzeitig erfassen zu können Werden unterschiedliche Hände erfasst ist es beinahe unmöglich ohne Verwendung eines Markers[2] zu wissen zu wem sie gehört. Werden optische Marker verwendet, so muss der Abstand und der Winkel der zwischen Kamera und Marker auftritt berücksichtigt werden. Die Kamera muss die Charakteristika eines Markers noch erkennen können (siehe Abbildung 4). Als günstig hat sich die Positionierung der Kameras an der Decke mit Blick auf die jeweilige Oberfläche erwiesen. Marker sind dann in der horizontalen Ebene anzubringen. Da das Licht von oben kommt ändert sich der Weiß- und Schwarzpunkt[3] nur geringfügig und die geometrische Verzerrung ist gering. [6]

Abbildung 4. Markierter Computer und Marker im Detail[6].

Ein großes Problem beim optischen Erkennungsverfahren besteht darin, dass das Kamerabild sehr stark von sich ändernden Lichtverhältnissen beeinflusst wird. Dieses Problem kann allerdings durch den zusätzlichen Einsatz von Infrarotkameras behoben werden. Allerdings würde der Einsatz von Infrarotkameras die Kosten der Interaktiven Oberfläche deutlich erhöhen. Ein weiters Problem der optischen Erkennung ist, dass durch den großen Rechenaufwand eine Echtzeitverarbeitung besonders schwer zu realisieren ist. Außerdem können in der Bildverarbeitung nur Merkmale extrahiert werden, die die Kamera auflösen kann. Im Vergleich zu anderen Verfahren wie z.B. kapazitive Erkennung ist die Auflösung der Kamera eher grob.

Eine andere Umsetzung der optischen Erkennung stellt Digital Vision Touch Technologie (DViT) dar. DViT ist eine kommerzielle Entwicklung im Bereich der optischen Erkennung von SMARTTech Inc. [7]. Bei dieser Technologie werden die Kameras nicht über der Interaktiven Oberfläche montiert, sondern in deren Ecken integriert. Die Position und der Abstand zur Oberfläche von einer oder mehreren Händen kann mittels moderner Bildverarbeitung aus dem Datenstrom ermittelt werden. Sind die Finger noch zusätzlich markiert kann auch zwischen verschiedenen Nutzern unterschieden werden.

Erkennung mittels Infrarot-LEDs

Dieses Erkennungsverfahren ist für speziell rechteckige Oberflächen geeignet. Zur Umsetzung dieser Technik werden an den Rändern der Interaktiven Oberfläche Infrarot-LEDs angebracht. An den jeweils gegenüberliegenden Seiten wird mittels einer Photodiode das Licht aufgenommen. Aus dem fehlenden Licht an 2 Photodioden kann die Position eines Objektes bestimmt werden. Der größte Nachteil dieses Verfahrens besteht darin, dass es bei der Benutzung mit mehreren Händen zu Verdeckungen kommen kann. Es lässt sich zwar die Position eines oder mehrere Objekte bestimmen, allerdings kann nicht zwischen den verschiedenen Objekten differenziert werden. Außerdem ist die Größer einer LED nach unten beschränkt was zur Folge hat, dass die Auflösung dieses Verfahrens eher gering ist [7]. Diese Art der Erkennung ist damit nur ziemlich eingeschränkt nutzbar.

Erkennung mittels Ultraschall

Um die Position eines Objektes im dreidimensionalen Raum exakt bestimmen zu können sind mindestens drei Ultraschallemitter notwendig. Das zu ermittelnde Objekt besitzt einen Ultraschallempfänger. Dieser Ultraschallempfänger kann die drei Emitter unterscheiden. Ist die Position der drei Sender bekannt kann die Position des Empfängers aus den Laufzeiten der Signale der 3 Emitter errechnet werden. [8]

Vorteile der Erkennung mittels Ultraschall sind die geringe Latenz der Positionsbestimmung, die Genauigkeit unter konstanten Rahmenbedingungen, die geringe Größe der Empfangskombination, dass Sender und Empfänger sich

[2] Marker sind klar erkennbare, orientierte Muster i.A. aus schwarzen und weißen Blöcken.

[3] Schwarzpunkt/Weißpunkt: Intensitätswert (beispielsweise in einem CCD), der als Schwarz oder Weiß interpretiert wird. Messwerte, die darunter oder darüber liegen werden uniform als Schwarz bzw. Weiß festgelegt. Damit wird der Helligkeitsumfang festgelegt.

optisch nicht "sehen" müssen und dass viele Objekte gleichzeitig beobachtet werden können. Nachteile sind der relativ kleine Erfassungsbereich und die Empfindlichkeit gegenüber Rahmenbedingungen wie Luftfeuchtigkeit und Lufttemperatur. Berücksichtigt werden muss bei der Umsetzung, dass Ultraschall Echos erzeugen kann. Der bedeutendste Nachteil bzw. Fehler der Ultraschalltechnik ist, dass der Benutzer einen Ultraschallemitter am Finger tragen muss, damit widerspricht diese Technik dem Grundsatz, dass sich der Benutzer weder vorkonfigurieren lassen muss noch zusätzliche Eingabengeräte benötigt.

Erkennung mittels Lasermessung

Aus einer Ecke der Projektionsebene wird ein rotierender Laserstrahl ausgesendet. Dieser deckt die Projektionsebene vollständig ab. Bei Kontakt mit der Hand wird ein Teil des Laserstrahls reflektiert. Aus dem Winkel der Reflexion und dem Anteil des Signals, das reflektiert wird lässt sich die Position der Hand bestimmen. Es ist günstig einen Laser im sichtbaren Spektrum zu verwenden. Der Benutzer kann dann sehen, wann er im sensitiven Bereich ist. Bei der Benutzung mit nur einer Hand kann die Position sehr genau bestimmt werden. Kommen mehrere Hände ins Spiel kann es wie auch bei der Erkennung mittels Infrarot-LEDs zu Verdeckungen kommen. [9]

Kapazitive Erkennung

Zu diesem Erkennungsverfahren existieren zwei Systeme: SmartSkin und DiamondTouch. Beide sind ähnlich aufgebaut und ihr Aufbau und ihre Funktionsweise wird im folgenden Teil genauer beschrieben.

SmartSkin [5]: dieses Verfahren beruht auf elektronischen Gesetzen. Unter einer dünnen Oberfläche liegt ein Geflecht leitender Kupferdrähte. Die eine Hälfte dieser Drähte ist parallel zueinander die anderen im 90 Grad Winkel dazu gespannt. Die waagrechten Drähte fungieren in diesem System als Empfänger und leiten das empfangene Signal an einen Konverter weiter, der sich in der Nähe der Interaktiven Oberfläche befindet. Die Drähte berühren sich nicht. Auf die senkrecht gespannten Drähte wird jeweils ein eindeutiges Referenzsignal (z.B. phasenverschobene Sinuswellen) gelegt. An den Kreuzungspunkten kommt es zwischen den Drähten zu kapazitiven Effekten. Was bedeutet, dass die in 90 Grad liegenden Drähte kapazitiv gekoppelt sind.

Bewegt sich ein leitfähiges Objekt, wie eine Hand über dem Drahtgitter, dann ändert sich das gekoppelte Signal. Aus der Änderung kann die Position der Hand bestimmt werden. Zur Funktionsweise siehe auch Abbildung 5.

Ein Vorteil dieser Erkennungsmethode besteht darin, dass die Position der Hand sofort erfasst wird, es gibt keine Latenz. Weiters lässt sich die Genauigkeit des Systems durch ein engeres Drahtgitter sehr leicht erhöhen. Und bei diesem Verfahren können auch mehrere Hände gleichzeitig erfasst werden.

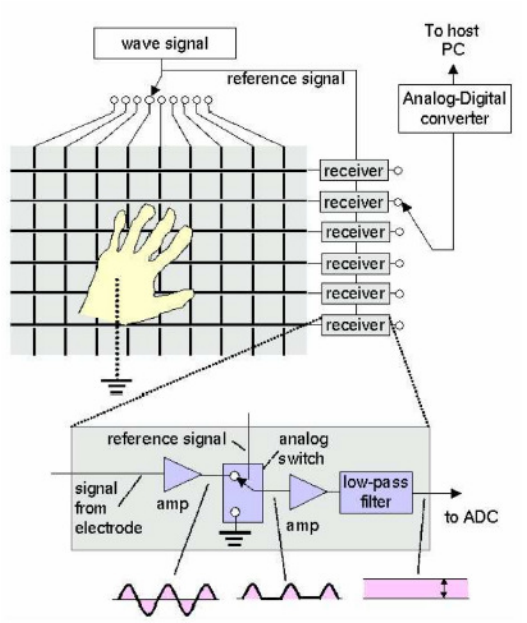


Abbildung 5. Physikalischer Aufbau von SmartSkin [5]

DiamondTouch [10]: In diesem System werden ebenfalls kapazitive Kopplungen verwendet, allerdings wird im Gegensatz zu SmartSkin für jeden "Punkt" der Interaktiven Oberfläche ein Erkennungselement verwendet. Als Erkennungselement werden Antennen, die mit unterschiedlich frequentem Strom belegt sind eingesetzt.

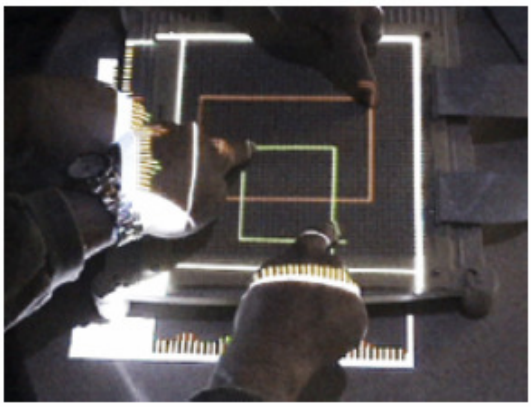


Abbildung 6. Zwei Benutzer bei der Benutzung von DiamondTouch [10]. An den verschiedenen Farben ist zu erkennen, dass zwei Benutzer unterschieden werden können.

Am Stuhl des Benutzers ist ein Empfangsgerät angeschlossen. Nähert sich ein Finger einer Antenne auf der Interaktiven Oberfläche, so beeinflusst das elektromagnetische Feld die Frequenz der Schwingung in Antenne und Empfangsgerät. Aus dem neuen Signal lässt sich bestimmen welcher Sender und Empfänger kapazitiv gekoppelt ist oder anders ausgedrückt welcher Benutzer gerade welchen Punkt der Oberfläche berührt. Damit ist es

auch bei diesem System möglich zwischen verschiedenen Benutzern zu unterscheiden (siehe Abbildung 6).

Beim Arbeiten auf der Interaktiven Oberfläche kann es natürlich auch vorkommen, dass ein Finger mehrere Antennen berührt. Doch auch in diesem Fall kann die Position genau ermittelt werden, da der Hauptberührungspunkt am stärksten im erzeugten Signal erkennbar ist.

Erkennung über Schall

Klopft man auf ein Material breiten sich die Schallwellen ringförmig von der Klopfstelle aus. Ist die Größe der Ebene auf die geklopft wird bekannt, kann über die an den vier Ecken positionierten Mikrophone die Position der Schallquelle und die Intensität des Schalls bestimmt werden. [9]

Bei dieser Art der Erkennung ist es so gut wie unmöglich eine Figur zu zeichnen, da nur das Auftreffen des Fingers auf der Oberfläche ermittelt werden kann. Die weiteren Bewegungen können nicht mehr verfolgt werden. Für dialogorientierte Systeme, wie die zu Beginn vorgestellte Autokonfigurations-Anwendung ist dieses Erkennungsverfahren allerdings sehr gut geeignet.

Diese Art der Erkennung ist auch besonders für Vandalismus-gefährdete oder dem Wetter ausgesetzte Installationen geeignet. Die Mikrophone sind hinter (einer dicken Glasscheibe) angebracht und das Projektionssystem kann an einer sicheren Stelle, abgekapselt angebracht werden. [1]

Qualitätsmerkmale eines Erkennungsverfahrens

Neben der genauen Ermittlung der Position einer Hand und der Orientierung greifbarer Objekte sind folgende Kriterien für die Güte einer Erkennungstechnologie wichtig: [13]

- Mehrere Berührungen können gleichzeitig erkannt werden;
- Der Kontaktpunkt kann einer Person zugeordnet werden;
- Objekte außerhalb der Interaktiven Oberfläche interferieren nicht mit der Erkennung;
- Die Erkennung ist unempfindlich gegenüber Umwelteinflüssen;
- Die Finger der Hand reichen zur Benutzung aus.
- Günstige Herstellung ist möglich.

Jun Rekimoto nennt in [5] zusätzlich die Erkennung des Abstands bzw. des Drucks auf die Interaktive Oberfläche.

Warum wird auf diese Merkmale Wert gelegt? Die Ermittlung der Position der Hand muss durchgeführt werden, damit bekannt ist wo der Benutzer gerade interagiert. Ist diese Position bekannt, so kann die Ausgabe darauf anpasst und beispielsweise ein Menü dargestellt werden. Die Orientierung greifbarer Objekte festzustellen ist notwendig um sie korrekt in die Interaktive Oberfläche einzubinden. Damit eine Interaktive Oberfläche realisiert werden kann, auf der mehrere Finger oder Personen gleichzeitig arbeiten können, müssen deren Kontaktpunkte gleichzeitig erkennbar sein. Die Zuordnung von einem oder mehreren Kontaktpunkten zu einer Person erlaubt es nachzuvollziehen wer was geändert hat und ermöglicht Sicherheitsmechanismen. Die Erkennung einzelner Finger ist wünschenswert damit auch kleinere Details auf einer Interaktiven Oberfläche geändert werden können und ermöglichen neue, schnellere Interaktionsmuster. Durch einen Schwellenabstand/-Druck kann ein Moduswechsel ausgelöst werden oder beispielsweise kann beim Zeichnen einer Linie die Dicke über den Druck variiert werden.

Die optische Erkennung, Erkennung mittels Ultraschall, Infrarot und Lasermessung können die Position der Hand oder anderer Objekte bestimmen, sie können aber nicht ohne Einbeziehung von Hilfskontrukten (Klicker, Gesten, Festlegen zweier Interaktionsebenen) erkennen, ob ein bestimmter Modus aktiv ist. Bei der Ermittlung der Position mittels Schall, kapazitiver Kopplung oder DiVT kann ein Moduswechsell durch Erhöhen des Drucks oder Verringern des Abstands erkannt werden. Diese Form der Interaktion ist sehr natürlich und somit intuitiv. Wenn der Benutzer auf ein Papier einen Strich zeichnen will muss er das Papier mit dem Stift berühren. Der Druck legt dann fest, wie der Strich aussieht. Diese Vorgehensweise lässt sich sehr gut auf Interaktive Oberflächen übertragen.

Erwähnenswert ist noch, dass aufgrund der horizontalen Orientierung der Interaktiven Oberflächen es sich anbietet diese mit greifbaren Objekten (tangible bits) zu kombinieren.

KOMBINATIONSMÖGLICHKEITEN UND PROBLEME DER TECHNIKEN

Um die speziellen Schwächen einzelner Sensortypen auszugleichen verwendet man oft eine Kombination verschiedener Typen, die sich gegenseitig ergänzen. Die entstehenden Systeme nennt man Hybrid-Systeme. Hybrid-Systeme sind heutzutage aus dem Tracking-Bereich kaum noch wegzudenken. Auch zu Redundanzzwecken werden verschiedene Sensoren kombiniert, um in Falle eines Ausfalls auf einen anderen Sensor zu wechseln.

Am offensichtlichsten ist die Aufwertung der kamerabasierten Technik durch den Einsatz eines Infrarotsystems. Durch die Kombination dieser beiden Techniken wäre die Möglichkeit gegeben eine Interaktive Oberfläche auch in nicht optimal belichteten Umfeldern einzusetzen.

Allerdings ist auch diese Kombination nicht an allen Orten verwendbar, da durch die direkte Bildübertragung der Kamera die Intimsphäre des Benutzers gestört werden könnte. Ein Beispiel hiezu wäre die Bedienungskonsole eines Multimediaplayers als Interaktive Oberfläche am Badewannenrand. Dieses Szenario ist allerdings mit keiner der bisher bekannten Techniken zur Umsetzung Interaktiver Oberflächen lösbar, denn der Einsatz von kapazitiver

Erkennung schließt sich in Feuchträumen selbst aus, die Erkennung mittels Infrarot-LEDs könnte nicht am Wannenrand realisiert werden und zur Ultraschallerkennung würde der Benutzer einen Ultraschallemitter am Finger benötigen, was in der Badewanne nicht empfehlenswert sein dürfte.

Das optische Erkennungsverfahren könnte durch den zusätzlichen Einsatz der Schallerkennung aufgewertet werden. Bei optischen Systemen kann bisher nicht wie bei kapazitiven Verfahren der Druck bzw. die Berührung des Fingers oder der Hand auf der Oberfläche festgestellt werden. Käme zusätzlich eine Schallerkennung zum Einsatz könnte der Benutzer Dinge auf der Interaktiven Oberfläche durch antippen auswählen. Dies würde Parallelen zu bereits bekannten, im alltäglichen Leben vorkommenden Aktivitäten aufweisen und so für den Benutzer leichter und intuitiv erlernbar sein.

Aus denselben Gründen wäre auch die Kombination des Infrarot-LEDs Erkennungsverfahrens bzw. des Laserverfahrens mit dem der Schallerkennung von Vorteil.

PROJEKTIONSSYSTEME

Wie bereits am Beginn des Papers erwähnt ist die dritte Komponente einer Interaktiven Oberfläche, neben der Interaktionsoberfläche und dem Erkennungsverfahren, das Projektionssystem. Grundsätzlich gibt es nur zwei verschiedene Technologien hierfür: Monitore und Projektoren. Da auf Interaktiven Oberflächen potenziell sehr informationsdichte Daten (z.B. Karten) dargestellt werden, sollte die Auflösung des Monitors oder des Projektors sehr hoch sein.

Selbstleuchtende Oberflächen (Monitore)

Monitore werden hauptsächlich in “Table-Top” Systemen eingesetzt. Dabei sitzt der Monitor in einer Halterung und die Bildfläche zeigt nach oben. Vorteil dieser Lösung ist, dass Monitore aufgrund der hohen Leuchtkraft auch im Freien eingesetzt werden können. Da ein Monitor aber eine festgelegte Bildschirmdiagonale hat, sind damit nicht beliebig große Flächen realisierbar.

Projektionsbasierte Darstellung

Für die Projektion werden handelsübliche Projektoren eingesetzt. Durch “Parallelschaltung” lassen sich damit beliebig große Flächen ausleuchten.

Grundsätzlich kann das Bild von vorne oder von hinten auf die Interaktive Oberfläche projiziert werden. Allerdings lässt sich die Rückprojektion nicht mit jeder Erkennungsmethode verwenden. Ein Beispiel dafür ist die kapazitive Erkennung. Die Entscheidung für Rück- oder Frontprojektion ist wichtig. Werden auf der Interaktiven Oberfläche greifbare Objekte eingesetzt und soll auf diese Objekte eine Grafik projiziert werden (beispielsweise, welche Musikspur gerade manipuliert wird [20]) dann muss eine Frontprojektion eingesetzt werden.

Die meisten Forschungsprojekte verwenden zur Darstellung LCD/DLP-Projektoren und Frontprojektion ein.

VERÖFFENTLICHTE PROJEKTE

In diesem Abschnitt werden zwei weitere Arbeiten im Bereich Interaktiver Oberflächen genauer vorgestellt.

Erweiterungen von Whiteboards

Whiteboards sind abwaschbare, hängende Oberflächen, auf denen mit speziellen Stiften geschrieben werden kann. Das auf der Tafel Notierte kann nur durch die Verwendung eines speziellen Schwamms wieder gelöscht werden. Das Whiteboard funktioniert also wie eine Tafel. Durch beispielsweise kapazitive Erkennung kann die Position von Stiften auf dem Whiteboard bestimmt werden.

Aufbauend auf das uns bekannte Whiteboard wurde das Smartboard [11] entwickelt. Dabei wird nicht mehr mit echter Tinte auf die weiße Oberfläche, sondern durch beobachtete Objekte (tracked objects) virtuell auf die Oberfläche gezeichnet. Ein beobachtetes Objekt kann in diesem Fall ein Stift sein, der malt und ein Quader der Inhalte löscht. Die Projektion wird dabei beispielsweise von einem LCD-Projektor übernommen. Dadurch lassen sich bestehende Computeroberflächen einbinden. Es ist zum Beispiel möglich eine Vorlesung mit PowerPoint zu halten und währenddessen mit dem interaktiven Stift Anmerkungen und Zeichnungen hinzuzufügen. Diese Daten können aufgezeichnet werden und den Studenten zur Verfügung gestellt werden. In der Vorlesung kann so deutlich flexibler auf Anregungen und Fragen aus dem Plenum eingegangen werden.

XEROX hat sich mit der Frage beschäftigt welche Interaktionsmöglichkeiten durch die neue Technik möglich werden. In [12] wird die Verwendung eines erweiterten Smartboards im Büro untersucht. Beispielsweise wird dem Benutzer immer Platz zur Eingabe zur Verfügung gestellt. Bestehende Inhalte werden verschoben und verkleinert, um Platz frei zu machen.

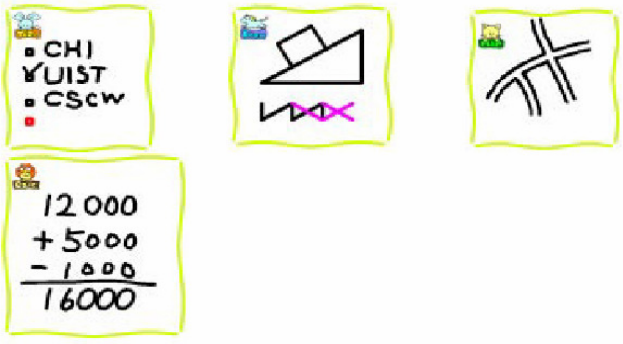


Abbildung 7. Verschiedene Arbeitsmodi auf dem Smartboard Flatlands von XEROX Parc [12].

Ein Beispiel zur Verwertung der Inhalte des Smartboards ist der eingebaute Rechner. Auf dem Smartboard werden der

Reihenfolge nach Auswahl, Zeichnung, Kartenerstellung und Rechnen Additionen und Subtraktionen handschriftlich untereinander geschrieben. Über OCR[4] werden die Summanden erkannt. Unter der Rechnung wird der Endbetrag eingeblendet. Dazu wird ein handschriftlicher Stil verwendet, um den informellen Charakter eines Whiteboards zu erhalten. Die Inhalte des Whiteboards können gesichert und wiederhergestellt werden.

Computerunterstütztes kooperatives Arbeiten

Die Mitarbeiter des Mitsubishi Electronic Research Laboratory machten sich darüber Gedanken, wie sich Interaktive Oberflächen für die Arbeit in (kleinen) Arbeitsgruppen einsetzen lassen Es wurden spontane Meetings beobachtet. Dabei stellte sich heraus, dass horizontale Oberflächen genutzt wurden um darauf Unterlagen zu legen und sie zu bearbeiten.

Die Positionierung des Dokuments legt dabei fest, ob es privat oder öffentlich ist. Wird ein Dokument in die Mitte gelegt wird dadurch impliziert, dass ein Kommentar der anderen Gruppenmitglieder erwünscht ist. Jedes Gruppenmitglied kann frei Änderungen und Vorschläge auf dem offen gelegten Dokument vornehmen. Ist hingegen das Dokument an der Kante des Tisches poitioniert und nur zu einer Person orientiert so ist implizit klar, dass der Inhalt "privat" ist.

Aus diesen Beobachtungen resultierte UbiTable. UbiTable hat die gesamte Anzeigenflächen in drei Kategorien gestuft: Der private Bereich, nämlich auf dem Bildschirm des Laptops bzw. PDAs, der persönliche Bereich und der öffentliche Bereich, die sich auf der Tischfläche befinden. Die letzten zwei unterscheiden sich in den Zugriffsrechten, im persönlichen Bereich kann der Gesprächpartner das Dokument lesen, aber besitzt keinerlei Kopien- oder Bearbeitenszugriffe. Wenn der Dokumentbesitzer das Dokument von sich fernhält bzw. zur anderen Person hinüber schiebt, dreht es sich in die Richtung des Gesprächpartners und ändert damit auch die Zugriffsrechte der anderen Person, so dass diese das Dokument bearbeiten oder zu sich nehmen kann.

Dieses System kann die Arbeit in Gruppen sehr positiv beeinflussen, da nicht jedes einzelnes Mitglied der Gruppe seine Vorschläge auf einem nur ihm ersichtlichen Exemplar des zu bearbeitenden Materials vornimmt und erst im Nachhinein ein Abgleich mit dem Rest der Gruppe vorgenommen werden kann. UbiTable bietet zwei große Vorteie: einerseits eine Zeitersparnis andererseits wird die Chance, dass Missverständnissen auftreten, minimiert. [3]

[4] Optical Character Recognition (OCR): Optische Schriftzeichen-Erkennung. Software, die gescannte Textdokumente oder Handschrift erkennen kann.

INTERAKTIONSMÖGLICHKEITEN MIT DER INTERAKTIVEN OBERFLÄCHE

Da Interaktive Oberflächen nicht mit Maus oder Tastatur gesteuert werden, sondern mit der Hand und den Fingern müssen für die Interaktion völlig neue Ansätze entwickelt werden. Es ist dabei schwierig und vielleicht auch gar nicht erwünscht auf die Paradigmen der Desktop-Welt zurück zu greifen. Vielmehr sollen Aktionen und Interaktionen aus dem Alltag übernommen werden, um dem Benutzer eine lange Eingewöhnungsphase zu ersparen.

Gesten

Von Geräten wie dem Palm Pilot oder dem Apple Newton ist die um Gesten erweiterte Benutzerführung bereits bekannt. Ein Strich nach links orientiert entspricht dem Löschen des letzten Zeichens, ein gegen den Uhrzeiger gerichteter Pfeil widerruft die letzte Aktion.

Der Benutzer ist im Allgemeinen mit einem Arbeitsablauf dieser Art und Weise nicht vertraut, daher ist in jedem Fall vor der Erstbenutzung dieses Systems eine Einarbeitungszeit notwendig. Aus diesem Grund sollte die Form der Gesten, so logisch und intuitiv wie möglich sein. Der Benutzer sollte nur Gesten benutzen müssen, die er aus dem realen Leben bereits kennt. Rekimoto hat hierzu einige neue Ideen bei der Verwendung Interaktiver Oberflächen entwickelt [5].

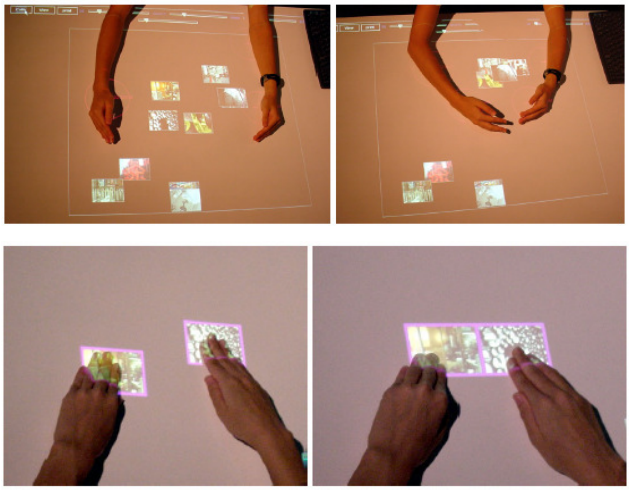


Abbildung 9. Geste zur Selektion von Objekten auf einer Interaktiven Oberfläche mit kapazitiver Erkennung [5]

Ein Beispiel dafür ist die Auswahl von Objekten durch "Zusammenraffen" (siehe Abbildung 9). Hierbei kann der Benutzer mehrere Objekte durch die Bewegung der Hände bzw. Arme auf einander zu auf einen Stapel zusammenschieben. Es wird also eine Geste imitiert, die der Benutzer aus dem Alltag kennt: Liegen mehrer Gegenstände nebeneinander kann man sie auch im realen Leben durch Zusammenschieben auf einen Stapel schieben. Dieses Vorgehen ist auch von Menschen, die noch nichts mit einem derartigen System zutun hatten ohne vorherige

Lernphase, allein durch eine „Try & Fail“ Herangehensweise nachvollziehbar.

Da mittels kapazitiver Erkennung nicht nur einzelne Hände sondern auch die einzelne Finger unterschieden werden können, eröffnen sich ganz neue Interaktionsmöglichkeiten:

Bei Manipulationen, bei denen zwei Werte (gleichzeitig) verändert werden, kann dies mittels zweier Finger simultan erfolgen. Ein Beispiel dafür ist die Manipulation zweier Ankerpunkte einer Bezierkurve oder Skalierung mit Drehung eines Bildes (siehe Abbildung 10).

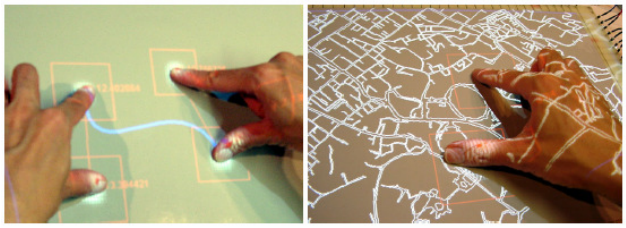


Abbildung 10. Gleichzeitige Veränderung zweier Ankerpunkte, rechts eine Bezierkurve links ein Kartenbrowsing System, auf einer Interaktiven Oberfläche mit kapazitiver Erkennung [5]

Des Weiteren ist es möglich nach dem auflegen der Hand auf die Interaktive Oberfläche an jeder der Fingerspitzen, die die Fläche berührt, ein Menüpunkt darzustellen. So ist es dem Benutzer möglich durch Bewegung der Einzelnen Fingerspitzen in der Menüstruktur zu navigieren (siehe Abbildung 11).

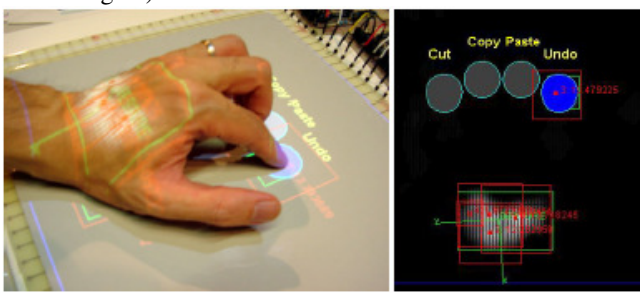


Abbildung 11. Fingergesteuerte Menüdarstellung auf einer Interaktiven Oberfläche mit kapazitiver Erkennung [5]

Greifbare Objekte (tangible bits)

Menschen sind an die Manipulation von greifbaren Objekten aus ihrem alltäglichen Leben gewohnt. Die Manipulation von greifbaren Objekten ist intuitiv und natürlich. Daher können greifbare Objekte in einer Benutzerschnittstelle sehr mächtige Werkzeuge sein. Im Kontext Interaktiver Oberflächen können Holzklötze unterschiedlicher Form auf Werkzeuge abgebildet werden. Ist einem Klotz eine Funktion, wie Rotieren oder Skalieren zugeordnet, dann muss noch festgelegt werden, unter welchen Bedingungen das Werkzeug aktiv ist. Durch Verwendung von Gesten im Umgang mit dem Klotz (Abdeckung des Klotzes durch die Hand, Greifen des Klotzes an der langen oder kurzen Seite, Abdecken des Klotzes mit beiden Händen) wird die Funktion des Klotzes mit einem virtuellen Objekt assoziert und im nächsten Schritt aktiviert. [20]

Eine andere Möglichkeit ist Daten mit einem greifbaren Objekt zu assoziieren. Im Szenario der Städteplanung könnte man jedem Gebäude einen Klotz zuordnen. Position und Orientierung im Raum des Gebäudes werden dann direkt über den Klotz festgelegt. Es stellt sich dann die Frage, ob auf die greifbaren Objekte gar nichts oder etwas spezielles projiziert werden soll. Um die Darstellung auf dem greifbaren Objekt synchron zu halten muss die Position, die Orientierung und die Form des greifbaren Objekts bekannt sein oder erkannt werden. Die Form des Objekts kann durch ein 3-D Modell bekannt gemacht werden. Mittels der Erkennung durch Ultraschall bzw. der optischen Erkennung lässt sich Position und Orientierung bestimmen. [21]

EINSATZGEBIETE UND ANWENDUNGSBEREICHE

Die für uns meistbekannte Möglichkeit ist die Touchscreenlösung, welche bei kleiner Fläche sehr gut einsetzbar ist. Es reagiert schnell auf die Aktion des Benutzers, und lässt sich immer günstiger produzieren. Es ist gut für kleine Geräte einsetzbar. Nachteile von Touchscreens, sind ihre Druckempfindlichkeit und Schwierigkeiten bei der Umsetzung auf Großflächen.

Da als Projektionsfläche z.B. eine feste Mauer verwendet und der Projektor an der Decke montiert werden kann, bieten sich Interaktive Oberflächen als Informationssysteme in öffentlichen Bereichen, die Vandalismusgefährdet sind an.

Ein weiteres Einsatzgebiet ist die Unterstützung von größeren Arbeitsgruppen. Beispielsweise können bei der Städteplanung [4] alle beteiligten Gruppen direkt am Prozess teilnehmen. Jeder kann Strassen verschieben, Gebäudehöhen festlegen. Der Rest der Beteiligten kann die Änderungen sofort wahrnehmen und kommentieren. Diese Änderungen können aufgezeichnet werden. Es kann mitprotokolliert werden, wer welche Veränderungen vorgenommen hat. Dieses Szenario kann also dazu beitragen Konsens herzustellen und nachzuvollziehen von wem welche Entscheidung getroffen wurde.

Ein weiterer Anwendungsbereich sind Spiele. Am MIT wurde ein Spiel entwickelt [14], bei dem zwei räumlich getrennte Spieler über eine Interaktive Oberfläche interagieren. Auf der Oberfläche werden virtuelle Kacheln dargestellt. Hinter den Kacheln ist ein Bild des anderen Spielers zu sehen. Mit einem Fußball wird auf die Interaktive Oberfläche geschossen. Die Intensität und die Position des Aufpralls werden mittels Mikrophonen gemessen. Übersteigt die Intensität in dem Areal der Kachel einen festgelegten Schwellenwert, so “bricht” sie. Ziel des Spieles ist es alle Kacheln des Gegners zu zerschießen.

Ein anderes Beispiel zur Anwendung Interaktiver Oberflächen ist iBar: iBar ist das System für die interaktive Gestaltung eines Bar-Tresen. Die milchige Bar-Oberfläche

kann mit eingebauten Beamern mit beliebigen Inhalten bespielt werden. Das integrierte Tracking-System erkennt alle Objekte, welche die Oberfläche der Bar berühren. Damit kann die Projektion auf die Position der Gegenstände reagieren oder projizierte Objekte können mit den Fingern angeklickt werden. Das iBar System lässt sich kompakt in die Theke einbauen, es benötigt keine weiteren Installationen an der Decke oder ähnlich. Die Software kann flexibel je nach Bedarf angepasst und erweitert werden. [18]

AUSBLICK

Interaktive Oberflächen werden bisher in klar definierten Konstellationen in genau definierten Anwendungsbereichen eingesetzt. Jedes Forschungssystem setzt auf andere Möglichkeiten der Eingabe und Darstellung. Es gibt nicht die von PCs bekannte Kombination von Fenstern, Maus, Tastatur und Icon. Jede Interaktive Oberfläche erfordert vom Benutzer neues Verstehen und Erlernen der Interaktion. Dieser Zustand ist unerwünscht. In der Zukunft werden sich Axiome der Benutzerinteraktion Interaktiver Oberflächen herauskristallisieren. Bisher ist die grafische Darstellung vielfach „hart codiert" in der Implementierung zu finden. Es gibt keine bestehenden Softwarekomponenten zum Aufbau von Interaktiven Oberflächen, keine GUI-Komponenten, die Events versenden etc. Eine vom Endbenutzer festlegbare Darstellung ist somit unmöglich. Es ist zu hoffen, dass es mittels Autorenwerkzeugen und Frameworks möglich sein wird, Interaktive Oberflächen stärker für den Endbenutzer zu individualisieren.

Es ist gut vorstellbar, dass in der Zukunft in den verschiedensten Bereichen des alltäglichen Lebens Interaktiven Oberflächen eingesetzt werden, sowohl zuhause oder im Büro, als auch unterwegs in Museen, auf der Strasse. Interaktive Oberflächen könnten für spezielle Zwecke, wie zum Beispiel zur Lebenserleichterung für behinderte oder gebrechliche, ältere Menschen oder auch Kinder eingesetzt werden.

Jede allein stehende Technologie ist weißt Lücken auf, das trifft auch auf Interaktive Oberflächen zu. In Kombination mit anderen Technologien könnten Interaktiven Oberflächen viel stärker in den Alltag integriert werden. Projektoren und Videokameras können immer kleiner hergestellt werden, daher wäre es möglich sie zum Beispiel in Brillen zu integrieren. Oder man könnte Interaktive Oberflächen mit Sprachenerkennungstools kombinieren um die Interaktionsmöglichkeiten zu erweitern.

CONCLUSION

In diesem Paper wurde gezeigt, dass einige Ansätze zur Umsetzung Interaktiver Oberflächen bereits existieren. Allerdings sind die meisten der vorgestellten Projekte lediglich Forschungsstudien und sind noch nicht alltagstauglich umgesetzt worden. Allerdings steht ohne Zweifel fest wie nützlich derartige Interaktionsmöglichkeiten speziell für nicht im vollen Umfang Funktions- oder Bewegungstüchtige Personen sein könnten. Aber auch im Arbeitsalltag wären Interaktive Oberflächen wie etwa der UbiTable für die Zusammenarbeit mit anderen Personen von Vorteil. Um dies zu realisieren und vor allem für die breite Masse zugänglich und bezahlbar zu machen ist aber noch einiges an Forschungsarbeit zu erledigen. Abschließend kann man sagen, dass Interaktive Oberflächen, bei richtigem Einsatz im Alltag eine Bereicherung darstellen.

LITERATUR

1. Paradiso, J., "Tracking contact and free gesture across large interactive surfaces", In: *Proc. Commun. ACM 2003, New York, USA,* (2003), pp 62-69.
2. Rekimoto, J., Saitoh, M., "Augmented surfaces: A spatially continous work space for hybrid computing environments", In: *Proc. SIG CHI 1999, Pittsburgh,: ACM* (1999).
3. Shen, C., "Ubitable: Impromptu face-to-face collaboration on horizontal interactive surfaces", In: *Proc. UbiComp 2003, Seattle, Washington, USA* (2003).
4. Ishii, H., Underkoffler, J., Chak, D., Piper, B., "Augmented urban planning workbench: Overlaying drawings, physical models and digital simulation", In: *Proc. IEEE & ACM ISMAR,* (2002).
5. Rekimoto, J. "Smartskin, An infrastructure for freehand manipulation on interactive surfaces", In: *Proc. SIG CHI 2002,* (2002).
6. Rekimoto, J., "Multiple-computer user interfaces: Beyond the desktop direct manipulation environments", In: *Proc. SIG CHI 2000,* (2000).
7. Smart Technologies Inc., "Dvit digital vision touch techologie White Paper", In: *http://smarttech.de/dvit/DViT_white_paper.pdf,* (12.06.2007).
8. Rolland, J., Davis, L., Baillot, Y., "A survey of tracking technology for virtual environments", In: *Fundamentals of Wearable Computers and Augmented Reality, Ed. Barfield and Caudell, Mahwah, NJ.*, (2001) pp 67–112.
9. Paradiso, J., Hsiao, K., Strickon, J., Lifton, J., Adler, A., "Sensor systems for interactive surfaces" In: *IBM Systems Journal 39* (2000) pp 892–914.
10. Dietz, P., Leigh, D., "Diamondtouch: A multi-user touch technology", In: *Proc. UIST 2001, Orlando, Florida USA*, (2001), pp 219-226.
11. Knowlton, N., "SMART Board Interactive Whiteboards – Built to Last", In: *http://www2.smarttech.com/NR/rdonlyres/1DD898C4-47D2-4BEB-8862-246DC964DE8C/0/2006SBiwBuilttoLastUSNPUpdated07.pdf*, (11.05.2007).

12. Mynatt, E.D., Igarashi, T., Edwards,W.K., LaMarca, A., "Flatland: New dimensions in office whiteboards", In: *SIGCHI, Pittsburgh, Pennsylvania, USA,* (1999), pp 346–353.

SEKUNDÄRLITERATUR

1. iBar – intelligent surface system, In: *http://www.i-bar.ch/info/*, (12.06.2007).
2. Mueller, F., Agamanolis, S., Picard, R.W., "Exertion interfaces for sports over a distance", In: *Proc. UIST 2002, Cambridge, USA*, (2002).
3. Xiaoping, Y, "Interaktive Oberflächen", http://www.vs.inf.ethz.ch/edu/SS2005/DS/reports/04-surfaces-report.pdf, (25.05.2007).
4. Ryall, K., Forlines, C., Shen, C., Morris, M.R., "Exploring the effects of group size and table size on interactions with tabletop shared-display groupware" In: *Proc. of the Conference ob Computer Supported Cooperative Work 2004,* (2004).
5. Bahr, J., „Interaktive Oberflächen & Table Top Benutzerschnittstellen", In: *http://www.hcilab.org/events/ui-update1.0/02-Interaktive_Oberflaechen-Julius_Bahr.pdf, (14.05.2007).*
6. Rauterberg, M., Fjeld, M., Krueger, H., Bichsel, M., Leonhardt, U., Meier, M., "Build-it: A planning tool for construction and design" In: Proc. SIGCHI 1998, New York: ACM Press, (1998), pp 177-178.
7. Ullmer, B., Ishii, H., "The metadesk: Models and prototypes for tangible user interfaces", In: *Proc. UIST, Cambridge, USA,* ACM (1997), pp 223–232.

Interface Design digitaler Bibliotheken für Kinder

Gabriele Pibal
Universität Klagenfurt
Austria
gpibal@edu.uni-klu.ac.at

ABSTRAKT

Kinder im Alter von 3 – 13 Jahren sind die größte Benutzergruppe in Bezug auf Computer oder Internet. Die Zunahme und Präsentation von Informationen jeglicher Provenienz via Internet ermöglicht einen schnellen Zugang zu diesen Informationen. Die Benutzeroberflächen digitaler Bibliotheken, die gerade eine gezielte Suche nach bestimmten Informationen unterstützen sollen, wurden allerdings für ältere Kinder, Studenten und Erwachsene entwickelt. Neben den Fragen der Realisierung von visuellen Suchsystemen wird die Differenz des Suchverhaltens von Kindern und Erwachsenen und deren Lösungsansätze exemplarisch anhand einer digitalen Tierbibliothek sowie der „International children's digital library" (ICDL) im Vergleich zu gängigen Online-Bibliotheken dargestellt.

Schlüsselwörter

Kinder, Interface digitaler Bibliotheken, Suchverhalten von Kindern im Internet, Repräsentation von Suchergebnissen, ZUIs (Zoomable User Interfaces).

EINLEITUNG

Bibliotheken als Ausdruck von „Wissensvermittlung" in Form von dokumentierter Information konnten aufgrund der exakten Aufzeichnungen und Klassifizierung der Bibliotheksbestände bereits sehr früh digital erfasst und durch Definition von Metastandards wie MAB (= Maschinelles Austauschformat für Bibliotheken) vernetzt werden. Im Vordergrund der Datenerfassung und dem Interface Design stand vor allem der schnelle und exakte Zugriff auf Informationen geordnet nach jenen Suchkriterien, die vom Benutzer eingegeben beziehungsweise ausgewählt werden konnten. Die Gestaltung der Benutzeroberfläche ist dementsprechend bis heute durch Texteingabefelder, Auswahl von speziellen Suchkriterien wie Autor, Titel, Jahr, Schlagwörter etc. zur Unterstützung des Suchverhaltens gekennzeichnet. Das Design der grafischen Oberfläche entspricht somit der Vorstellung des Suchverhaltens von älteren Kindern und Erwachsenen.

Generell wird in der Forschungsliteratur zu diesem Thema die mangelnde Benutzerzentrierung als Defizit bei der Entwicklung von visuellen Suchsystemen angesehen. [10] So ist etwa die Unterteilung von Welt in Begriffen via Klassifizierungen, Thesauren oder Ontologien aus Sicht der Entwickler sinnvoll, entspricht aber nicht unbedingt der Sichtweise der Bibliotheksbenutzer. Die globale Einteilung von verfügbarem Wissen ist für die Benutzer meist zweitrangig, wesentlich ist vielmehr, dass die Suche mit den von ihnen geläufigen Suchbegriffen erfolgen kann. [6]

METAPHER BUCH ALS VISUALISIERUNGSKOMPONENTE

Für die Darstellung von Klassifizierungshierarchien Informationen betreffend wurden in der Vergangenheit für Kinderbibliotheken meist vorgetäuschte Bücher als Metapher verwendet.[1] Dies ist vor allem darauf zurückzuführen, dass Bücher für uns – egal wie alt wir sind – nach bestimmten Kriterien ausgewählt werden, sei es in einer Buchhandlung oder Bibliothek. Gerade im Zeitalter Digitaler Bibliotheken hat die Präsenz-Bibliothek keinesfalls an Bedeutung verloren. Im Gegenteil: Das Auffinden eines Buches im Bücherregal, gekennzeichnet durch den Gang zwischen Bücherregalen als adäquate Strategie der Literatursuche, kann als „Urahn aller Browsing-Systeme" angesehen werden. [6]

Nachfolgend sei hier als Beispiel für die Verwendung der Metapher „Buch" als Visualisierungskomponente die Online-Bibliothek für Studierende der University of California, Los Angeles (UCLA) angeführt.

[1] Als bekanntestes Beispiel dafür wird in der Literatur der Science Library Catalog (SDL) angeführt. Mitte der 90er Jahre entwickelt, benötigte dieses System bereits keinen Keyword-Input mehr. [Vgl.3, S. 398]

Smithells Metals Reference Book
* Looking for a specific journal article? Use Citation Linker
* E-books In Science and Engineering!
* Tips for Graduate Level Research
* Publishing and Teaching in the Digital Age
Engineering
RSS-New Acquisitions Beta Service
Science
Find Articles in Journals and Conference Papers (Guides)
Click and select
Find Books/E-books, Journals/E-journals, and more (Guides)
Click and select

Abbildung 1. Science & Engineering Library (SEL) [2]

Neben anderen Designkomponenten wie Texteingabefelder zur Suche von wissenschaftlichen Artikeln werden verschiedene "Bücher" in einer Slide Show angezeigt. Klickt man auf ein Buch, so gelangt man via Verlinkung zu einem im anglo-amerikanischen Raum bekannten Portal, dass darauf spezialisiert ist wissenschaftliche Werke im Bereich Software Engineering, Physik, Mathematik etc. zu verkaufen. Allerdings wird auch ein freier Zugang zu den Werken speziell für Studenten angeboten. Die Selbstdefinition dieser Online-Bibliothek hebt vor allem neben der Möglichkeit der Analyse und der Reorganisation der Daten in über 800 Werken (Papers, Handbücher etc.) die Optionen des Sortierens, Filterns und Exportierens der Daten hervor.

In einem Bericht der *School of Library and Information Science*, verfasst von David V. Loertscher und Blanche Woolls [9] werden unter dem Punkt "Information Literacy in the World of Electronic Information" die von Mark von Wodtke definierten „Gedankenprinzipien" zitiert[3]:

> Mind over Media (Gedanken über Medien)
>
> Menschen können mehr lernen als nur einen Button anzuklicken. Durch den Gebaruch von elektronischen Medien können sie lernen interaktiv zu arbeiten.
>
> Navigating (Navigation)
>
> Menschen können das Navigieren durch Informationsumgebungen lernen.
>
> Mapping Media Space ("Landkarten" von Medienräumen)
>
> Menschen können kognitiv "Landkarten" von Medienräumen entwickeln – die Informationsumgebung, in der sie arbeiten.
>
> Visualization (Visualisierung)
>
> Menschen können Modelle der Visualisierung jener Realitäten entwickeln, mit denen sie arbeiten.
>
> Creative Thinking (Kreatives Denken)
>
> Menschen können lernen ihr kreatives Potential zu nutzen, wenn sie mit elektronischen Medien arbeiten.
>
> Collaboration (Zusammenarbeit)
>
> Menschen können in den entstehenden elektronischen Informationsumgebungen zusammen lernen und arbeiten.
>
> A Mind Primer ("Gedankenzünder")
>
> Studenten können zu einem sehr frühen Zeitpunkt lernen ihre kreativen gedanklichen Fähigkeiten zu entwickeln.

Gerade Wodtkes letztes "Gedankenprinzip" scheint UCLA umgesetzt zu haben, indem sie ihren Studenten über die Verlinkung von der Studentenbibliothek zur Knovel Library das so genannte vernetzte Denken in der Literaturrecherche ermöglicht.[4]

Im europäischen Raum sind vor allem institutionelle Bibliotheken miteinander vernetzt und ermöglichen so den Zugriff auf größere Datenbestände. Die Verlinkung mit Online-Buchhandlungen ist eher unüblich und wenn es Verlinkungen gibt, dann sind diese auf den jeweiligen Intstitutshomepages von Universitäten, je nach Ausrichtung der wissenschaftlichen Disziplinen, zu finden.

ONLINE BIBLIOTHEKEN

Generell sind Online-Bibliotheken vor allem im universitären Bereich so gestaltet, dass der Benutzer seine Suchanfrage via Keywords eingibt und über boolsche Operatoren (UND, ODER, NICHT) die jeweilige Suchanfrage eingrenzen kann. Weiters wird das Suchverhalten der Benutzer in „Suche" und „Expertensuche" unterteilt. Für den deutschsprachigen Raum wird meistens das OPAC System (= Online Public Access Catalog), ein digitaler Bibliothekskatalog, verwendet und die Benutzeroberfläche der jeweiligen Bibliothek mittels Logo angepasst. Da dieser Katalog Einträge von Buchtitel, Verfasser, Ort, Jahr,

[2] http://www.library.ucla.edu/sel/.

[3] Vgl. von Wodtke, Mark. Mind over Media: Creative Thinking Skills for Electronic Media. New York: McGraw-Hill, 1993, zitiert nach [9].

[4] Vgl. http://www.knovel.com

Verlag, Schlagwörter, ISBN, Systematisierungsnummer, Aufstellungsnummer etc. zulässt, stellen diese Daten die Auswahlmöglichkeit in den Datenfeldern dar.

UNIVERSITÄT
KLAGENFURT
Universitätsbibliothek
Anmelden | Neustart
Suche | Ergebnisliste
Einfache Suche | Suche mit Filter
Einfache Suche
Suchbegriff(e)
Datenfeld
Alle Datenfelder
OK
Verwerfen
Suchtipps:
Wenn Sie die Option Phrasensuche wählen, wird das System nur Titel finden, in denen
Sie brauchen Groß- und Kleinschreibung nicht zu beachten.
Zwischen den Wörtern wird eine logische UND-Verknüpfung angenommen. Sie können
Das Fragezeichen ? wird zur Trunkierung verwendet. Mit der Eingabe von Arch? finden S
Sie können dieses Trunkierungszeichen auch verwenden, um unterschiedlichen Schreib
(Schiffahrt) als auch die neue Schreibweise (Schifffahrt) zu finden.

Abbildung 2. Universitätsbibliothek Klagenfurt [5]

Die Suchergebnisse werden in einer Listenform ausgegeben, die mit verschiedenen Funktionalitäten ausgestattet ist. Optionen wie das Drucken der Suchergebnisse und Abfragen nach Angaben zum Standort in der Bibliothek, der Verfügbarkeit, der Entlehndauer etc. sind für den Benutzer möglich. Dies ist im Prinzip der Ausdruck eines hierarchischen Ordnungssystems, der von Bibliothekaren definiert wurde um Bücher systematisch erfassen zu können, erweitert mit benutzerfreundlichen Features, die die Abläufe in einer Bibliothek für den Benutzer vereinfachen sollen.

So sind digitale Bibliotheken für uns immer mit einer realen Bibliothek verbunden, d.h. nach Ausgabe der Suchergebnisse können wir auf aufgrund der Angaben direkt auf das gesuchte Werk zugreifen. Wesentlich dabei ist, dass Benutzer anhand der Gestaltung des Buches eine Zuordnung zum Verlag herstellen und somit eine erste Relevanzentscheidung treffen können. Ebenso stellen Umfang und Art des Buches einen wichtigen Faktor bei der Informationssuche dar. [6]

Es verwundert daher nicht, dass die Internationale Jugendbibliothek in München ebenfalls das OPAC System für die Bestandserfassung von Kinderbüchern verwendet, allerdings auch für die Benutzersuche. Benutzerzentriertes Interface Design wird hier nur mittels eines animierten Rundgangs erzeugt, indem durch das Anklicken des Bibliotheksgebäudes mit der Maus Bilder eingespielt werden, die den Blick in das innere des Gebäudes freigeben.

Abbildung 3. Rundgang[6]

Das Suchverhalten von Kindern ist gegenüber Erwachsenen vor allem in der noch nicht ausgeprägten Fähigkeit der Abstraktion bestimmt, d.h. sie suchen intuitiv. Kinder können mit den o.a. hierarchischen Suchsystemen nicht interagieren, da eine direkte Eingabe von Keywords für Grundschulkinder eine oft unlösbare Aufgabe darstellt. Bezüglich der Suchanfragenoptimierung über logische Verknüpfungen ist bekannt, dass Kinder erst ab dem 8 – 10 Lebensjahr die Fähigkeit entwickeln in Abstraktionen zu denken. Die Eingabe von mehreren Keywords als Suchanfrage, die logisch miteinander verknüpft werden müssen um ein exaktes Suchergebnis zu liefern, stellen ein dementsprechendes Hindernis dar. [8]

Das Fehlen der Benutzerzentrierung bei der Entwicklung von visuellen Suchsystemen wird deshalb auch darauf zurückgeführt, dass die meisten Systeme zwar im Rahmen von Forschungsprojekten entwickelt wurden, aber die technische Realisierung der Visualisierungsidee im Vordergrund stand, nicht die Benutzerinteressen. [10]

INTERFACE DESIGN TIERBIBLIOTHEK

Der Versuch einer optimalen grafischen Auflösung von Suchkriterien wurde beim Design einer Tierbibliothek für Kinder durch die Miteinbeziehung der zukünftigen Benutzer, eben den Kindern, in einem Forschungsprojekt realisiert.

A priori stellt sich bei der Entwicklung von Bibliotheken, unter Einbeziehung der Benutzerzentrierung, für Kinder die Frage: Wie eignen sich Kinder Wissen an? Diese Fra-

[5] http://opac.uni-klu.ac.at/

[6] http://www.ijb.de

gestellung war Ausgangspunkt der Entwicklung des Interfaces der Tierbibliothek.

Eine wichtige Rolle spielte ebenso das Wissen, dass Kinder Bilder, Videos oder die Stimmen der von ihnen gesuchten Tiere sehen bzw. hören wollen. Der Designprozess wurde daher als Partnerschaft definiert, d.h. Entwickler lernen von Kindern, die das Expertenwissen bezüglich "Usability"[7] haben, wie sie als Experten dieses Wissen technisch realisieren sollen.

Als Usergruppe wurden Kinder im Alter von 5 - 10 Jahren gewählt. Das Projekt umfasste ein interdisziplinäres Team von Software Entwicklern, Pädagogen und Kindern, wobei den Kindern die Rollen von Designern sowie Informanten im Forschungsprozess zugeschrieben wurden um die Benutzerzentriertheit des Interface Designs zu gewährleisten.

Der Designprozess wurde in folgende Phasen unterteilt:[8]

(i) Brainstorming

(ii) Analyse bereits existierender digitaler Tierbibliotheken

(iii) Testen

Die Phase des Brainstormings ermöglicht hier den Erwachsenen den Zugang zu den Anforderungen und Vorstellungen einer Bibliothek der zukünftigen Benutzer.

In der nächsten Projektphase, der Analysephase, wurden die Ergebnisse der Analyse schriftlich erfasst und ausgewertet. Die Auswertung ergab, dass sowohl Kinder wie Erwachsene stark animierte Oberflächen ablehnten. Dies lässt den Schluss zu, dass unabhängig vom Alter bestimmte Designelemente als störend bei der Informationssuche und der damit verbundenen Wissensaneignung empfunden werden. In diesem Fall jene Elemente, die das gezielte Auffinden von bestimmten Informationen durch zu starke grafische Animationen verlangsamen oder den Benutzer vom Suchen an sich abhalten.

Beim Testen ergab sich unter anderem die Idee der Unterteilung des Screens in verschiedene Bereiche, d.h. man stellte fest, dass Tiere eigentlich an verschieden „Orten" zu finden sind, die eben grafisch dargestellt werden sollten. Der Start-Screen wurde daher in „Lebensräume" unterteilt:

(i) Zoo

(ii) Welt

(iii) Suche

Für die Analyse des Suchverhaltens wurden User-Tracking Tools eingesetzt, deren Analyse unter dem Aspekt der Verweildauer Differenzen des Suchverhaltens von Mädchen und Jungen ergaben. Mädchen neigen eher zum Browsen, Jungen wollen nach einer gestellten Suchanfrage schneller das Ergebnis erhalten. Somit ergab sich, dass beide Strukturen, Browsen wie Suchen, unterstützt werden mussten.

Wesentlich für die Kinder war, dass die Tierbibliothek variantenreich bezüglich Hintergrunddarstellung und Designvielfalt sein sollte. [3] Da das „Sehen" bei Kindern eine zentrale Rolle spielt, wurden ZUIs für das Anzeigen der Suchergebnisse entwickelt und eingesetzt, damit die Kinder ihre "Tierwelten", die die Weltsicht der Kinder widerspiegeln, im Detail betrachten konnten.

ASSOZIATIVES SUCHVERHALTEN

Die Differenzierung beim Interfacedesign bezüglich des Alters ergibt sich im Hinblick auf „Usability" im Bereich der Suchanfrageerstellung, d.h. hierarchisches versus assoziatives Suchverhalten. Visualisierte Darstellungsformen werden immer öfter für assoziativ verbundene Konzepte eingesetzt, damit der Benutzer die verschiedenen Abbildungsvorgänge interaktiv beeinflussen kann. [10]

Speziell für Kinder sind visualisierte Darstellungsformen daher besonders geeignet, ebenso ist aber auch die Art der Visualisierung im Hinblick auf die Feinmotorik entsprechend dem Alter abzustimmen. Bei der Tierbibliothek beinhaltet das Interface Design daher große, leicht klickbare Icons.

Bezüglich des assoziativen Suchverhaltens wurden die Suchkategorien so definiert, dass Grundschulkinder sowohl Simple-Search wie Boolean-Search leicht handhaben können, indem sie die entsprechenden Items nacheinander anklicken. Die Reihenfolge des Klickens definiert die Suchart und ermöglicht logische Verknüpfungen. Da das Suchverhalten intuitiv erfolgt, benötigen Kinder eindeutige Symbole die die möglichen Fragekriterien darstellen. Deshalb wurde bei der Tierbibliothek die Differenzierung zwischen medialen oder hierarchischen Fragekriterien mittels zweier Figuren, den „QueryKids", realisiert. Die Funktionsbeschreibung wird nachfolgend anhand der Suchmöglichkeiten dargestellt.

Den Beginn der Suche zeigt Abbildung 4, die Kinder können hier taxonomisch suchen, indem sie je nach Informationsinteresse die Auswahl zwischen den Suchbereichen Zoo oder Welt treffen.

[7] In der Literatur gibt es derzeit keine eindeutige Definition von „Usability", am häufigsten wird damit eine Benutzerfreundlichkeit bezeichnet die hier übernommen wird.

[8] Unter Testen wird hier die permanente Weiterentwicklung verstanden, bis das Ergebnis für alle Beteiligten als zufrieden stellend angesehen wurde. In der Literatur wird diese Form auch als „Agile Modeling" bezeichnet.

Abbildung 4. Start-Screen[9]

Abbildung 5 zeigt jenen Suchbereich, der Informationen über die Tiere durch Auswahl von Kategorien ermöglicht:

- Was essen die Tiere?
- Wie leben die Tiere?
- Wie bewegen sich die Tiere? usw.

Abbildung 5. Suche

Hier sind die „QueryKids“ Dana und Kyle zu sehen. Beim Anklicken von Medien-Items wird Kyle angezoomt, hierarchische Items zoomen zu Dana. Die jeweiligen Such-Items bei Kyle oder Dana repräsentieren visuell die formulierten Fragen. [3]

In Abbildung 6 können die Tiere im Zoo über die Suchkategorie Haus wie das Vogelhaus, oder das Reptilienhaus, gefunden werden. Allerdings bleibt hier die Frage offen, ob die Darstellung für eine Tierbibliothek geeignet ist, da ein Zoo an sich die Problematik der artgerechten Tierhaltung beinhaltet.

Abbildung 6. Zoo

In der „Weltsicht“ (Abbildung 7) können Kinder geographisch nach Tieren suchen. Mit einem Mausklick auf die ausgewählte Region werden die Tiere angezeigt. Durch das Anklicken eines spezifischen Tieres können erfasste Bild-, Audio- und Videodaten angezeigt werden, ebenso Informationen in Form von Texten, je nach Interesse und Informationsbedarf.

Abbildung 7. Welt

Der Suchprozess nach Bildern von „fliegenden Tieren“ die „Pflanzen essen“ würde wie folgt ablaufen: [4]

1. Klick des Items, das Bilder repräsentiert.
2. Klick der Kategorie „Wie bewegen sich die Tiere?“[10]
3. Klick des Items “Fliegen”.
4. Navigation in die Hierarchie: Suchanfrage nach Bildern von fliegenden Tieren.

[9] Abbildung 4 – 7 wie die Funktionsbeschreibung wurden von [13], Research Projekt Screenshots entnommen.

[10] Hier ist die Kamera bei Kyle, die Vorschaubilder im Ergebnisbereich.

Abbildung 8. Wie bewegen sich Tiere[11]

5. Klick der Kategorie “Was essen die Tiere?”
6. Klick auf “Pflanzenesser”, damit ist die Suchanfrage abgeschlossen.
7. Klick auf den Ergebnisbereich, damit können die Suchergebnisse angesehen werden.

Die Anwendung konstruiert automatisch eine „Vereinigung“ von jedem Term in der selben Top-Level Hierarchie sowie den „Durchschnitt“ bei divergierenden Top-Level-Hierarchien. So würde beim Anklicken der Icons Fisch, Vogel und „essen Fleisch“ automatisch folgende Abfrage generiert:

((Fisch ODER Vogel) UND „essen Fleisch“) [4]

Somit können Kinder mit diesem Interface einfache boolesche Verknüpfungen erzeugen, ohne deren mathematische Bedeutung zu kennen und ihre Suchanfragen nach Informationsinteresse eindeutig definieren.

INTERFACE DESIGN ICDL

Basierend auf dem Erfolg der Tierbibliothek wurde das Interface Design der ICDL ähnlich gestaltet. Die digitale Bibliothek enthält rund 800 Kinderbücher und wird von ca. 25.000 Benutzern aus 150 Staaten genutzt. Das ICDL Projekt umfasste ebenfalls ein interdisziplinäres Team von Software Entwicklern, Informationsexperten, Pädagogen und Künstlern, sowie Kindern im Alter von 6 – 11 Jahren.

Die erste ICDL Version bestand aus 2 Interfaces, die den Zugang zu rund 500 Büchern in 30 Sprachversionen ermöglichte. Der Internetzugang wurde mit einer Java Applikation, entsprechendem Plugin, einer Breitbandverbindung sowie einem HTML Interface für ein 56K Modem realisiert. [8]

Die hierarchische Suchstruktur bestand ursprünglich aus 14 Kategorien im Top-Level Bereich. [6]

Hier wurden die Sucharten Simple-Search und Boolean-Search anhand der Forschungsergebnisse der Tierbibliothek umgesetzt. Ebenso wurde die Anzeige der Suchergebnisse via ZUIs realisiert.

Abbildung 9. Original ICDL Version, Kategorien

Die neue Version des Start-Screens wurde mit runden Grafikelementen realisiert, da aus Platzmangel die rechteckige Darstellung der Suchkategorien eine Einschränkung derselben beinhaltete. Um eine Überladung und damit eine Überflutung des Benutzers von Informationsinhalten zu erreichen, wurde die Kategoriestruktur vereinfacht. Nach Auswertung von Weblogdaten[12], die das Suchverhalten der Kinder als User-Profiles wiedergeben, wurden die Kategorien nach Auswahlhäufigkeit definiert. Weiters wurden runde Icons für Kategorien und rechteckige Icons für Bücher verwendet, da Kinder diese Differenzierung zur Unterscheidung von Suchkategorien und Büchern benötigen. [7]

Nachdem das gesuchte Buch als Suchergebnis angezeigt wird, haben die Benutzer die Möglichkeit die Anzeige der Bücher auszuwählen und somit die grafische Darstellungsform selbst zu bestimmen, in der sie das Buch lesen wollen. Hier wurden drei grafische Modi der Ansicht realisiert:

(i) Normalansicht

(ii) Comic

(iii) Spirale

[11] Abbildung 8 wurde aus [3] entnommen.

[12] Zeitraum der Daten: 1 Jahr.

Diese Sichten werden in den nachfolgenden Abbildungen dargestellt.[13]

Abbildung 10. Lesesichten

Sucht man Kinderbücher nach den Kriterien „3 – 5 Jahre“, „alle Sprachen“ und „Farbe des Buches“, so können nach der Auswahl aus den Suchergebnissen die Buchseiten via Mausbewegung vergrößert angezeigt werden, d.h. hier wird das Bilderbuch für Kinder, die noch nicht lesen können, in seiner klassischen Form durch den Einsatz von ZUI Applikationen visuell umgesetzt.

Ein Ziel der ICDL ist unter anderem bei Kindern durch den Bibliotheksbestand das Interesse an anderen Kulturen zu wecken. Deshalb gibt es eine Plattform zur Bewertung der gelesenen Bücher, wo Benutzer Rezensionen der von ihnen gelesenen Bücher „posten“ können und so ihre Erfahrungen anderen zugänglich machen. Gleichzeitig wird damit eine weitere Interaktionsmöglichkeit angeboten, die des interkulturellen Lernens im Netz.

So sehr diese digitale Bibliothek auch genutzt wird, sie ersetzt aber keinesfalls Bücher als Drucksorten, d.h. die digitale und die traditionelle „Buchwelt“ werden von den Kindern als gleichwertig angesehen. Dies hängt vor allem mit den verschieden Erlebnismöglichkeiten zusammen, die das Buch in digitaler oder gedruckter Form kennzeichnen. Die Druckversion beinhaltet vor allem haptische Dimensionen, die digitale Version die des Browsens und Suchens.

ZUIS

Bei visualisierten Darstellungsformen im Bibliotheksbereich ist es essentiell, dass die Suchergebnisse im Detail angesehen werden können und ZUIs ermöglichen die Unterstützung der Detailansicht. Gerade Kinder im Grundschulalter eignen sich Informationen in erster Line visuell an.

Das Projekt Tierbibliothek inkludiert die Entwicklung eines Tools für Kinder um die digitale Bibliothek testen zu können. Dabei wurde auf eine bereits erfolgreich eingesetzte Entwicklungsumgebung zurückgegriffen, die für ein anderes Projekt verwendet wurde, nämlich die *KidPad* Architektur, die mit einer Zoominging Applikation ausgestattet war. *KidPad* wurde unter anderem für das „Geschichtenerzählen“ von Kindern erfolgreich eingesetzt[14] und basiert auf *Jazz* und *MID,* beides Open Source Java Werkzeuge. *Jazz* unterstützt ZUIs durch das Erzeugen eines hierarchischen Scenegraphen für 2D Grafiken, MID unterstützt multiple Input Devices für Java. [2] Hinter dem „Geschichtenerzählen“ steht die Idee, dass Kinder ihre Geschichten zum Leben erwecken können, indem sie zwischen gezeichneten Elementen, die digital erfasst wurden, zoomen.

Bei der ICDL wurde daher vor allem *Jazz* weiterentwickelt, da die grafische Darstellung der Bücher, je nach Anzahl der Suchergebnisse, durch die Bildschirmgröße begrenzt ist. Weiters erfordert die Leseansicht das Zoomen der Bücher um das Lesen ermöglichen zu können.[15]

RESÜMEE

Kinder erwarten sich offensichtlich von neuen Technologien mehr Spaß und diesem Faktor wird durch die Realisierung der Auswahl der o.a. grafischen Modi Raum gegeben. Der Thematik des “Spaß Habens am Tun” wird in der Literatur ebenso bezüglich des Designs von Interfaces für Erwachsene immer mehr Bedeutung beigemessen und ist nicht nur mehr auf Kinder beschränkt. [11]

Die Ausgabe der Suchergebnisse in Listenform scheint nicht mehr das einzig adäquate Mittel der Darstellung zu sein, da Informationen multimedial in vielfältiger Form darstellbar sind. Die in der Arbeit exemplarisch vorgestellten Interfaces, speziell für Kinder entwickelt, zeigen Lösungsansätze im Bereich visualisierter Darstellungsformen von Suchergebnissen im Bibliotheksbereich., die vor allem durch die benutzerzentrierte Ausrichtung bei der Entwicklung entstanden sind. Speziell am Beispiel der Tierbibliothek wurde gezeigt, dass die Visualisierungswünsche das Interface betreffend von Kindern und Erwachsenen gleich waren. Derzeit scheint die Realisierung einer Bibliothek in Form der ICDL unrealistisch ob des immensen Arbeitsaufwandes einen Bibliotheksbestand

[13] Abbildung 10 wurde von [13], Research Projekt Screenshots entnommen.

[14] Eine Projektkooperation von Nottingham University (UK), Royal Institute of Technology (KTH Stockholm), University of Maryland (USA), Swedish Institute of Computer Science (SICS, Stockholm).

[15] Der Quellcode von *KidPad* kann unter [12] bezogen werden.

derart digital zu erfassen. Allerdings könnten kleine Buchbestände in Schulbibliotheken oder Kinderbücher von Online-Buchhandlungen ähnlich der ICDL visuell aufbereitet werden.

LITERATURVERZEICHNIS

1. Bederson, B., Grosjean, J. and Meyer, J. Toolkit Design for Interactive Structured Graphics. In: IEEE Transactions on Software Engineering, Vol. 30, No. 8, August 2004 1, pp. 1-12.

2. Benford, S. et. al. Designing Storytelling Technologies to Encourage Collaboration Between Young Children. CID-69, KTH, Stockholm, Sweden 2000. Paper accepted to CHI-2000, the Hague, the Netherlands, 1-6 April, 2000, pp. 1-11.

3. Druin, A. et. al. Designing a digital library for young children. In: *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries 2001*, Roanoke, Virginia, USA, ACM Press New York, 2001, pp. 398 – 405.

4. Druin, A. et. al. A Collaborative Digital Library for Children: A Descriptive Study of Children's Collaborative Behavior and Dialogue. In: Journal of Computer-Assisted Learning, 19 (2), 2002, pp. 239-248.

5. Druin, A. et. al. Children's interests and concerns when using the International Children's Digital Library: A four country case study, HCIL-2007-02, http://cgis.cs.umd.edu/hcil/pubs/tech-reports.shtml, 2007-04-22.

6. Hanke, P., Mandl, T., Womser-Hacker C. Ein „Virtuelles Bibliotheksregal" für die Informations-wissenschaft als Anwendungsfall semantischer Heterogenität, In: Hammwöhner, R., Wolff, C., Womser-Hacker, C.: Information und Mobilität, Optimierung und Vermeidung von Mobilität durch Information. *Proceedings des 8. Internationalen Symposiums für Informationswissenschaft (ISI 2002)*, Regensburg, 8. – 11. Oktober 2002, Konstanz: UVK Verlagsgesellschaft mbH, 2002, S. 289 – 302.

7. Hutchinson, H., Bederson, B. and Druin, A. The Evolution of the International Children's Digital Library Searching and Browsing Interface. In: *Proceedings of the 2006 Conference on Interaction Design and Children IDC '06*, Tampere, Finnland, ACM Press New York, 2006, pp. 105-112.

8. Hutchinson, H. et. al. How do I find blue books about dogs? The errors and frustrations of young digital library users. In: *Proceedings of HCII 2005,* Las Vegas, NV (CD-ROM). HCIL-2007-02, http://cgis.cs.umd.edu/hcil/pubs/tech-reports.shtml, 2007-04-22.

9. Loertscher, D., Woolls, B. The Information Literacy Movement of the School Library Media Field: a Preliminary Summary of the Research, School of Library and Information Science San Jose State University, San Jose, California, USA, http://witloof.sjsu.edu/courses/250.loertscher/modelloer.html, 2007-05-12.

10. Reiterer, H. Visuelle Recherchesysteme zur Unterstützung der Wissensverarbeitung. In: Hammwöhner, R., Rittberger, M., Semar, W.: Wissen in Aktion. Der Primat der Pragmatik als Motto der Konstanzer Informationswissenschaft, Konstanz: UVK. Schriften zur Informationswissenschaft Band 41, 2004, S. 1-22.

11. Shneiderman, B. Designing For Fun: How to Make User Interfaces More Fun. In: ACM Interactions 11, 5 (Sept-Oct 2004), pp. 48-50.

INTERNET

12. http://www.cs.umd.edu/hcil/, 2007-05-09
13. http://www.childrenslibrary.org, 2007-05-09
14. http://www.ijb.de, 2007-06-14
15. http://www.knovel.com, 2007-05-09
16. http://www.library.ucla.edu/, 2007-05-09
17. http://www.uni-klu.ac.at/ub/, 2007-05-09